

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах**

До захисту допущено:
Завідувач кафедри
_____ Олександр РОЛІК
«__» _____ 20__ р.

**Дипломний проєкт
на здобуття ступеня бакалавра
за освітньо-професійною програмою «Програмне забезпечення
інформаційно-комунікаційних систем»
спеціальності 121 «Інженерія програмного забезпечення»
на тему: «Система керування мобільними роботами»**

Виконав (-ла):
студент (-ка) IV курсу, групи ІТ-61
Александров Данііл Сергійович _____

Керівник:
Доцент каф. АУТС, к.т.н., доц.
Катін Павло Юрійович _____

Рецензент:
С.н.с. ІТГП НАН України, д.т.н.,
Биченок Микола Миколайович _____

Засвідчую, що у цьому дипломному
проєкті немає запозичень з праць інших
авторів без відповідних посилань.
Студент (-ка) _____

Київ – 2020 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма «Програмне забезпечення інформаційно-комунікаційних систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 20__ р.

ЗАВДАННЯ

на дипломний проєкт студенту

Александрову Даніїлу Сергійовичу

1. Тема проєкту «Система керування мобільними роботами», керівник проєкту Катін Павло Юрійович кандидат технічних наук, доцент, затверджені наказом по університету від 07 травня 2020 р. №1081-с

2. Термін подання студентом проєкту 09.06.2020

3. Вихідні дані до проєкту

Мова програмування C/C++, фреймворк ROS, середовище програмування Visual Studio Code.

4. Зміст пояснювальної записки

1. Вступ 2. Аналіз предметної області 3. Аналіз існуючих рішень 4. Розроблення системи керування мобільної платформи

5. Перелік графічного матеріалу

1. Діаграма активності

2. Алгоритм побудови шляху

3. Діаграма розгортання

4. Діаграма прецедентів

6. Дата видачі завдання «3» березня 2020 року

Календарний план

№ з/п	Назва етапів виконання Дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Дослідження актуальності теми	17.03.20	Виконано
2	Дослідження об'єктної моделі системи	19.04.2020	Виконано
3	Огляд існуючих рішень системи	27.04.2020	Виконано
4	Розробка моделі архітектури системи	04.05.2020	Виконано
5	Дослідження інструментів для реалізації системи	15.05.2020	Виконано
6	Реалізація модулів системи	29.05.2020	Виконано
7	Оформлення пояснювальної записки	20.05.2020	Виконано
8	Подання готового проєкту	09.06.2020	Виконано

Студент

Даніїл АЛЕКСАНДРОВ

Керівник

Павло КАТІН

АНОТАЦІЯ

Александров Д. С. Система керування мобільними роботами. КПІ ім. Ігоря Сікорського, Київ, 2020.

Проект містить 64 с. тексту, 28 рисунків, 9 таблиць, 18 посилань на літературні джерела, додатки 4 кресленика

Ключові слова: навігація, мобільні роботи, побудова маршруту, визначення положення на місцевості, тріангуляція.

Об'єктом розробки є система керування для мобільних роботів.

Мета розробки – знайти оптимальне рішення для покращення способу навігації та керування мобільних роботів.

У дипломному проекті досліджено недоліки різних способів побудови навігації та запропонований власний спосіб для уникнення відомих помилок. Проведено ретельне дослідження інструментів, для прискорення розробки системи. Досить багато уваги було приділено дослідженню відомих способів навігації та вивчення їх недоліків. Так як для керування мобільними роботами, система навігації обирається з огляду на той факт де та в яких умовах буде використовуватись система, судячи з цих даних проводиться дослідження характеристик місцевості на які можна зав'язати контрольні точки.

Отримані результати допоможуть спростити та покращити керування і навігацію для мобільних роботів, при використанні в середовищі з обмеженою кількістю орієнтирів.

SUMMARY

Daniel Aleksandrov, Control system of mobile robots. KPI of Sikorsky name, Kyiv 2020.

The project contains 64 pages, 28 figures, 9 table, links to 18 literary sources and 4 design schemas.

Keywords: navigation, mobile robots, path planning, localization, triangulation.

The object of development is the system of navigation creation and control for mobile robots.

The purpose of the development is to find the optimal solution to improve the way mobile robots navigate.

The diploma project explores the shortcomings of different ways of constructing navigation and proposes its own way to avoid known errors. A thorough study of the tools to accelerate the development of the system. quite a lot of attention was paid to the study of known methods of navigation and the study of their shortcomings. Since the navigation system is chosen to control mobile works based on the fact where and under what conditions the system will be used, based on these data, a study of the characteristics of the area to which you can set checkpoints.

The results will help simplify and improve navigation and navigation for mobile robots when used in a environment with a limited number of landmarks..

Поз.	Формат	Позначення	Найменування	Кількість в аркуші	№ прим.	Примітки
1			<u>Документація загальна</u>			
2						
3			Знову розроблена			
4						
5	A4	IT61.020БАК.004 ПЗ	Система керування мобільними роботами.	1		
			Пояснювальна записка			
6	A3	IT61.020БАК.004 Д1	Система керування мобільними роботами.	1		
			Діаграма діяльності			
7	A3	IT61.020БАК.004 Д2	Система керування мобільними роботами.	1		
			Алгоритм побудови маршруту			
8	A3	IT61.020БАК.004 Д3	Система керування мобільними роботами.	1		
			Діаграма розгортання			
9	A3	IT61.020БАК.004 Д4	Система керування мобільними роботами.	1		
			Діаграма прецедентів			

Змін.	Лист	№ докум.	Підпис	Дат
Розроб.	Александров Д.С.			
Перевір.	Катін П. Ю.			
Н. Контр.				
Затверд.	Ролік О.І.			

IT61.020БАК.004 ТП

Система керування мобільними роботами. Відомість технічного проекту	Лім.	Лист	Листів
		1	1

КПІ ім. Ігоря Сікорського
кафедра АУТС гр. IT-61

Пояснювальна записка
до дипломного проєкту
на тему: «Система керування мобільними
роботами»

Київ – 2020 року

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ.....	3
ВСТУП.....	4
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	6
Висновок до розділу.....	16
2 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ.....	17
2.1 Лінійна навігація.....	17
2.1 Навігаційна система з використанням одометрії.....	21
Висновок до розділу.....	27
3 РОЗРОБЛЕННЯ СИСТЕМИ КЕРУВАННЯ МОБІЛЬНОЇ ПЛАТФОРМИ..	28
3.1 Дослідження інструментів для реалізації проекту.....	28
3.1.1 Архітектура ROS.....	30
3.2 Хід виконання проекту.....	33
3.2.1 Розробка діаграми активності.....	33
3.2.2 Розробка алгоритму побудови маршруту.....	34
3.2.3 Розробка діаграми прецедентів системи.....	35
3.2.4 Побудова діаграми розгортання.....	42
3.2.5 Реалізація системи керування.....	43
Висновок до розділу.....	61
ВИСНОВКИ.....	62
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	63

				ІТ61.020БАК.004 ПЗ					
аркис	№ докум.	Підпис	Дата						
Розробив	Александров Д.С.			Система керування мобільними роботами.			Літ.	Лист	Листів
Перевір.	Катін П.Ю.								2
Реценз.				Пояснювальна записка			КПІ ім. Ігоря Сікорського кафедра АУТС гр. ІТ-61		
Н. контр.									
Затверд.	Ролік О.І.								

ПЕРЕЛІК СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

АКП – Автоматично Керований Пристрій

САПР – Система Автоматизованого Проектування

AMR – Autonomous Mobile Robot

DSP – Digital Signal Processor

EKF – Extended Kalman Filter

GPS – Global Positioning System

IMU – Inertial Measurement Unit

RF – Radio Frequency

ROS – Robotic Operating System

RTOS – Real-Time Operating System

SAS – Side-Angle-Side

SLAM – Simultaneous Localization and Mapping

ToF – Time of Flight

TCP – Transmission Control Protocol

					ІТ61020БАК.004 ПЗ	Архив
Зм.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

Від початку свого існування люди намагаються різними способами автоматизувати свою роботу. В наш час роботизовані системи знаходять широке застосування в різних сферах життєдіяльності людини. Вони працюють на заводах виконуючи тяжку роботу по збиранню автомобілів або ж займаються сортуванням нашої пошти. Всю рутинну роботу можна доручити роботам, а люди хай займаються більш важливішими справами.

Так як роботи вже починають своє існування поруч з людьми я б хотів розкрити тему про їхнє переміщення та орієнтування на місцевості, як вони це можуть робити, яким чином вони розуміють де вони знаходяться та куди їм потрібно рухатись.

Для будь-яких мобільних роботів, орієнтування на місцевості невід'ємна частина їхнього існування. Уникнення небезпечних ситуацій, таких як зіткнення з перешкодами та ризикованих умов (високі або низькі температури, потрапляння у воду чи радіація), але при розробці роботів враховуються умови в яких система буде використовуватись та враховуються можливі ситуації. В даній роботі будуть розкриті способи керування та побудови навігації для роботизованих систем, типи навігації, висвітлені їхні недоліки та запропоновані шляхи їх вирішення.

Навігація роботизованих систем включає в себе здатність визначення власного положення відносно стартового положення та побудова шляху до заданої точки в навколишньому середовищі, для цього їм потрібна мапа цього середовища. Тому на даний час це дуже актуальна тема дослідження, для розвитку сфери розробки роботизованих систем, яка скоро увійде в нашу буденність та буде допомагати в нашому повсякденному житті.

Об'єктом дослідження даного дипломного проєкту є розроблення системи керування мобільними роботами, побудова навігації та створення роботизованої платформи.

					IT61020БАК.004 ПЗ	Арквш
Зм	Арк.	№ докum.	Підпис	Дата		4

Предметом даного проєкту є мобільна платформа яка включає в себе систему керування та використання навігації на місцевості.

Мета даного проєкту полягає в тому, щоб дослідити та розробити систему, яка може працювати на різних пристроях та буде легка в допрацюванні.

Засобом досягнення поставленої мети є розробка мобільної платформи та системи керування, яка спрощує розробку та дасть можливість дослідити недоліки в розробці подібних систем для уникнення їх в майбутньому, це дозволить скоротити час розробки подібних систем наявністю готового базового модуля.

Тому тема бакалаврського проєкту, яка націлена на дослідження та розробку системи керування мобільними роботами, на даний момент є актуальною.

Дипломний проєкт складається з наступних розділів: вступ, аналіз області дослідження, огляд існуючих рішень, розробляння системи керування мобільної платформи, висновок та перелік використаних джерел.

					IT61020БАК.004 ПЗ	Аркvш
Зм	Арк.	№ докvм.	Підпис	Дата		5

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Навігація – область науки, яка вивчає процес моніторингу та контролю переміщення об’єктів з однієї точки в іншу. Навігація складається з чотирьох головних категорій: наземна навігація, морська навігація, аеронавігація та космічна навігація. Для роботизованих систем це невід’ємна частина їхнього програмного забезпечення, адже навігація грає велику роль під час керування роботом, це «очі» робота з допомогою навігації приймаються рішення, що до побудови шляху до точки в просторі, рішення для запобігання зіткнень з перешкодами. Система повинна мати достатню кількість сенсорів для отримання інформації про навколишнє середовище, щоб виконувати корисні дії, принцип роботи системи зображений на (рис. 1.1).



Рисунок 1.1 - Замкнена кільцева система

За таким принципом система отримує інформацію з датчиків, в залежності від змін в навколишньому середовищі приймає рішення на подальші дії [1].

В більшості випадків керування роботом покладено на плечі оператора безпосередньо на рівні пересування робота, за для такого керування оператору

					IT61020БАК.004 ПЗ	Архив
Зм	Арк.	№ докum.	Підпис	Дата		6

потрібний безперервний зоровий контакт з системою та моментальне коригування його дій. Такий підхід накладає велику кількість обмежень, виключає автономну роботу робота та самостійне прийняття рішень.

В перших навігаційних системах для роботів за основу брались скануючі датчики, локаційні та стерео далекоміри. Спеціальна схема обчислення робота в кінці кінців приводила електричні сигнали до аналогів різних перешкод та приймала рішення на подальший рух.

Зазвичай навігація робота поділяється на три рівня, відповідно ближній, середній та дальній.

Навігація робота на дальньому рівні призначена для планування основного маршруту для робота. Головною функцією комп'ютерного зору є розпізнавання орієнтирів, Оптично-електрична схема яка забезпечує рішення даної задачі, складається з об'єктива зі змінною фокусною відстанню, електричного блоку який керує камерою механізму нахилу та повороту камери. Вхідні сигнали визначаються грубою картою видимості, візуальними моделями орієнтирів, картою місцевості та описом завдання. Уявлення про зовнішнє середовище базуються на карті областей видимості (прохідності робота), місцезнаходження робота, послідовності розташування областей, через які проходить маршрут руху.

На середньому рівні навігація містить карту, яка є підмножиною карти системи дальньої навігації, яка містить більш детальну інформацію. Завдання навігації полягає в забезпеченні руху в межах однорідної видимості, тобто робот проходить коридори вільного простору де не потрібно маневрування. Система проміжної навігації передбачає чергування таких коридорів і їх послідовне коригування шляхом збільшення ширини і розбиття маршруту на більш дрібні ділянки. Вхідні дані цієї системи будуються на мапі дальньої навігації, моделях відомих перешкод та очевидних орієнтирів місцевості, маршруті, спланованому на базі системи дальньої навігації.

					IT61020БАК.004 ПЗ	Арквш
Зм	Арк.	№ доквм.	Підпис	Дата		7

Навігація ближнього рівня використовується для безпосереднього вимірювання відстаней в поєднанні з визначенням проміжної безпечної зони, в межах якої рухається робот, а також аналізу структури місцевості. Вхідними даними служать інформація, яка надходить від модулів обчислення пройденого шляху та курсу, відомості про вільний простір. Система повинна вимірювати відстані, оцінювати структуру місцевості, визначати безпечний обхід перешкод і планувати проходження по певним траєкторіям.

Окремим завданням системи ближньої навігації є слідування маршруту. До її складу входять планування послідовності значних змін маршруту, подолання пересічених і викривлених ділянок, крутих спусків і підйомів дороги, а також, забезпечення навігації при наявності інших об'єктів. Таким чином, ця задача основна для всієї навігації робота та була пов'язана з першими етапами розробки навігаційних систем мобільних роботів.

Процес керування під час навігації робота полягає в передачі завдань від рівнів з вищим ступенем абстракції до нижчих рівнів, а інформація про стан робота проходить в зворотному напрямку. При цьому кожен рівень навігації зберігає мапу своєї робочої зони [2].

Навігація мобільних роботів охоплює великий діапазон різних технологій і застосувань. Вона спирається як на дуже старі технології, так і на самі новітні досягнення науки і техніки.

Перші моделі промислових більш-менш автономних роботів, були створені в 60-і роки, вони пересувалися по маршруту, заданому маршруту за допомогою електричних кабелів, прокладених під підлогою заводських споруд. На роботах були встановлені нескладні датчики прийому електромагнітного випромінювання кабелю, що дозволяли визначати де проложений кабель та слідувати йому. Апарати могли рухатися по різних маршрутах завдяки тому, що за кількома кабелям передавався сигнал з різною частотою. Але така схема була дорогою і негнучкою.

					IT61020БАК.004 ПЗ	Арквш
Зм	Арк.	№ доквм.	Підпис	Дата		8

Коли з'явилися перші системи комп'ютерного зору, вдалося відмовитися від кабелів, та використовувати камери, які розпізнавали лінії нанесені на підлогу, таким чином робот самостійно рухався вздовж цих ліній. Правда, лінії часто стиралися, нерідко перекривалися іншими апаратами і людьми, а в місцях де лінії перетиналися, роботи зазвичай втрачали здатність на подальший рух і зупинялися.

Також перевірялися інші схожі концепції. Маршрут руху будувався з предметів маркерів певної форми, які розміщувались на певній висоті, робот за допомогою простих датчиків розпізнавав ці маркери, визначаючи таким чином своє місцезнаходження. Але така система навігації заснована на небажаному фізично активному контакті машини з навколишнім середовищем, що може обернутися руйнівними наслідками. Крім того, роботи не завжди могли правильно розпізнавати маркери, а їх розташування доводилося вибирати дуже точно.

Поступово моделі навігації з використанням маркерів були обладнані більш досконаліми аналоговими датчиками, навчилися вимірювати силу реакції контакту і визначати форму маркера, а зараз в цих цілях застосовуються цифрові матричні датчики, здатні одержувати від маркерів докладні дані про навколишнє середовище.

Інший спосіб навігації використовує лазерні далекоміри та ультразвукові датчики. Однак лазерний промінь ефективний тільки в зоні прямої видимості. Крім того, на шляху променю часто виникають дрібні перешкоди, що вносять похибку в такий спосіб. Ультразвукові датчики мають значний недолік у вигляді великого часу відгуку (якщо робот знаходиться на великому і відкритому просторі), порядку десятих часток секунди, що обмежує швидкість роботи. До того ж швидкість звуку в різних умовах відрізняється, впливаючи на точність оцінки відстані, в результаті робот спотворює загальну картину навколишнього середовища. Створення тривимірних карт за допомогою лазерів в масштабі реального часу ще більш

					ІТ61020БАК.004 ПЗ	Арк.вш
Зм	Арк.	№ док.м.	Підпис	Дата		9

складніше і як мінімум, вимагає значних обчислювальних потужностей, які поки не вдається втілити в вигляді компактних бортових плат. З цих причин інформація отримана від бортових датчиків має невелику цінність. Роботу необхідно перевести її в формальний і структурований опис світу [4].

Одним із способів організації руху робота в заздалегідь не певному середовищі може бути використання алгоритмів системи управління рухом робота, забезпеченого оптопарою - датчиком стеження за смугою, нанесеною на поверхню полігону. Автономне визначення на борту робота його узагальнених координат, дозволяє сформувати сигнал пропорційний відхиленню робота від віртуальної смуги [5].

Сьогодні, більшість роботів, використовують за основу навігаційної системи одометрію (odometry - вимір пройденого шляху) для орієнтування на місцевості. Звичайний для отримання одометрії використовуються оптичні енкодери, розміщені безпосередньо на осях коліс.

Перелік обертальних датчиків, які використовуються сьогодні для вимірювання швидкості та переміщення роботів:

- енкодери зі щітковими контактами;
- потенціометри;
- оптичні енкодери;
- магнітні енкодери;
- електромагнітні енкодери.

Найчастіше зустрічаються обертальні інкрементні або абсолютні оптичні енкодери.

Сучасні оптичні датчики складаються з зменшеного сенсору, який фіксує переривання променя. У ньому сфокусований і спрямований на певний фотодетектор промінь світла періодично переривається диском зі спеціальними прорізами, що обертається на валу.

Існує два основних види оптичних енкодерів:

- інкрементний – вимірює швидкість обертання;

					ІТ61020БАК.004 ПЗ	Арквш
Зм	Арк.	№ докum.	Підпис	Дата		10

- абсолютний – вимірює точне кутове положення і може визначити швидкість.

Зазвичай абсолютні енкодери застосовуються в системах з повільним обертанням, для яких категорично не припустима втрата інформації по причині втрати енергопостачання. Найкраще підходить для систем з повільним і або нечастим обертанням, в яких кодується кут повороту (протилежність обчисленням, пов'язаних з тривалим високошвидкісним обертанням, які використовуються для обчислення пройденого шляху).

Головний недолік абсолютного енкодера - паралелізм вихідних даних, який вимагає більш складний інтерфейс і складається з великої кількості проводів.

Короткочасно одометрія дає хорошу точність, до того ж вона недорога і має дуже велику частоту дискретизації. Але за основу одометрії взято накопичення даних про положення робота, які збільшуються під час руху, що неминуче призводить до накопичення помилки. На практиці, накопичення помилки орієнтації є головною причиною більшості помилок позиціонування, кількість яких зростає пропорційно шляху, пройденого роботом.

Однак одометрія грає велику роль в навігаційній системі робота і завдання навігації спростилося, якщо б точність одометрії була збільшена.

Нижче подана низка причин, через які одометрія використовується в мобільних роботизованих системах:

- одометричні дані можуть бути об'єднані технологією абсолютного позиціонування (і іншими технологіями) для отримання кращої і більш точної оцінки положення;

- одометрія може бути використана в абсолютному позиціонуванні, покращеному орієнтирами (маяками) на місцевості. Таким чином точність одометрії буде підвищена, що дозволить зменшити частоту оновлень в абсолютному позиціонуванні;

					IT61020БАК.004 ПЗ	Арквщ
Зм	Арк.	№ доквм.	Підпис	Дата		11

- в деяких випадках одометрія використовується тільки в якості навігаційного інформатора.

Також існує альтернативний метод одометрії - інерційна навігація. Принцип роботи полягає в безперервному зчитуванні навіть найменших прискорень по кожній з трьох осей напрямків і переміщення в часі, щоб отримати і положення. Гіроскоп стабілізує платформу з сенсором, це необхідно для збереження суворої орієнтації трьох акселерометрів протягом всього процесу.

Концепція цього методу проста, але специфіка реалізації досить вимоглива. Головним чином це викликано помилками, причиною яких є відсутність стабільності гіроскопів, які використовуються для забезпечення коректності обчислень положення.

Однією з переваг інерційної навігаційної системи можна виділити можливість швидко фіксувати, ледь помітні динамічні дані.

Головним недоліком вважається те, що дані швидкості та куту повороту використовуються два рази, відповідно для обчислення орієнтації і лінійного позиціонування.

Інший спосіб для навігації використовує орієнтири. Орієнтири можуть бути штучні та природні.

Природні орієнтири мають головну проблему яка полягає у визначенні та зіставленні характерних особливостей сенсорних даних. З таких сенсорів складається комп'ютерний зір. Зазвичай системи комп'ютерного зору працюють на розпізнавання довгих відрізків прямих, наприклад коридорів, точок з'єднання стін та стелі.

Найбільш простішим способом є виділення штучних орієнтирів, так як вони створюються з оптимальними характеристиками для спрощення розпізнавання. Таким чином можна заздалегідь задати системі особливі характеристики орієнтира. Більшість систем використовують комп'ютерний зір для позиціонування по штучним орієнтирам, в якості орієнтирів

					IT61020БАК.004 ПЗ	Аркvш
Зм	Арк.	№ докum.	Підпис	Дата		12

найчастіше використовуються чорний прямокутник з білими крапками по кутах, а для калібрування системи використовують сферу з вертикальними і горизонтальними колами, це дозволяє визначити положення в просторі по одному зображенню.

Точність цього методу залежить від того, як добре будуть розпізнані геометричні параметри орієнтирів в сцені, під час отримання цих даних велику роль грає положення і кут робота відносно орієнтира.

Також існують інші орієнтири, для розпізнавання яких не потрібні візуальні сенсори. Серед таких орієнтирів широкого застосування отримали штрих-код відбивачі, які розпізнаються з допомогою лазерних сканерів.

Лінійна навігація також використовує орієнтири для позиціонування. Даний спосіб навігації можна виділити, як навігація по безперервним орієнтирам, але такий вид має дуже значний недолік в тому, що датчик фіксування орієнтиру повинен знаходитись в безпосередній близькості до лінії маршруту. Даний тип навігації зазнав широкого застосування на довгі роки в задачах промислової автоматизації, пристрої які працювали в такий спосіб зазвичай називали Автоматично Керовані Пристрої. Однак, такий вид навігації був не достатньо детально вивчений, в наслідок цього спосіб виключає самостійну побудову маршрутів переміщення робота.

Основні способи для реалізації лінійної навігації:

- електромагнітне керування;
- керування відбиваючою або оптичною стрічкою;
- керування магнітною стрічкою, де використовується феріто-магнітне напилення;
- керування по термальним маркерами.

Головні особливості навігації по орієнтирах:

- навігація по природним маякам вимоглива до стабільності оточення;
- навігація по штучним маяках – недорога та може включати додаткову інформацію;

					ІТ61020БАК.004 ПЗ	Арквш
Зм	Арк.	№ докum.	Підпис	Дата		13

- максимальна відстань між роботом і орієнтиром значно менше, ніж в системах з активними маяками;

- відстань та кут між роботом і орієнтиром впливає на точність позиціонування;

- необхідна велика обчислювальна потужність, ніж в системах з активними маяками;

- погодні умови та зміна часу доби можуть бути причиною помилок;

- в навігації по орієнтирах роботу потрібно приблизно знати своє початкове положення для того, щоб він знав де шукати орієнтири:

- потрібно знати розташування всіх маяків в просторі.

Широкого застосування зазнала навігація по карті. Картографічне позиціонування також відоме як зіставлення карти, такий вид навігації вимагає використання датчиків для сканування навколишнього середовища, для побудови локальної карти місцевості. Після того як локальна карта побудована вона порівнюється з глобальною, яка попередньо завантажена в пам'яті робота. Система зіставляє ці дві карти вираховує збіги та визначає своє поточне місцеположення та орієнтацію на місцевості. У вигляді початкової карти місцевості може виступати САПР модель місцевості, або вона може бути сформована при першому скануванні місцевості.

Побудова карти має проблему, яка дуже тісно пов'язана з можливостями сприйняття, сенсори отримують інформацію про навколишнє середовище з різних положень і цю інформацію потрібно правильно опрацювати.

Відображення, що використовується для побудови карти, повинне забезпечувати можливість оновлення карти новою інформацією, що надходить від сенсорів. Також необхідно отримувати актуальну інформацію для планування маршруту та запобігання зіткнень з перешкодами.

Виділяються три основні етапи в обробці даних для побудови карти:

- отримання характерних ознак з необроблених даних;

- комбінування даних від різних сенсорів;

					ІТ61020БАК.004 ПЗ	Архив
Зм	Арк.	№ доквм.	Підпис	Дата		14

- автоматична побудова абстрактної моделі місцевості.

Найбільш енергозатратний аспект картографічної навігації зіставлення карт. Порівняння відбувається при первісному отриманні характерних ознак, потім визначається точна відповідність між зображенням і характеристиками моделі мапи. В сфері комп'ютерного зору робота найчастіше фокусується на проблемі порівняння зображень, отриманих з випадкових положень та орієнтації відносно початкової моделі мапи.

Алгоритми порівняння даних поділяються на алгоритми які аналізують зображення та ті які аналізують характерні ознаки. Перший тип алгоритму простіший, тому що зіставити дані інформативної точки на місцевості простіше, ніж дані що представляють собою невеликий набір особливостей.

В алгоритмах, які засновані на аналізі особливих характеристик обчислення відбуваються швидше, ніж в алгоритмах які засновані на аналізі зображень і не потребують попередніх обчислень. Обчислення в алгоритмах заснованих на аналізі зображень можуть виконуватися з меншою кількістю даних, ніж в алгоритмах аналізу за характерними ознаками та можуть працювати з не ідеальною моделлю місцевості та видавати більш точні.

Подібно до навігації по орієнтирах, доцільніше використовувати приблизні координати положення робота, які отримані з одометрії, для отримання приблизної карти місцевості при вже відомій початковій карті по якій буде рухатись робот. Після цього, відбувається зіставлення даних отриманої карти з даними які робот отримує на даний момент. Даний підхід значно зменшує час для знаходження відповідностей.

В системах позиціонування, які засновані на аналізі характерних ознак, існує досить значна проблема, вона полягає в тому, що робот може не розпізнати об'єкти, які знаходяться досить близько до нього. На практиці ця проблема стає досить значною, особливо якщо для розпізнавання характерних ознак використовуються датчики, які мають досить малий кут огляду, тим самим обмежуючи отримання достатньої кількості інформації.

						ІТ61020БАК.004 ПЗ	Арквш
Зм	Арк.	№ докум.	Підпис	Дата			15

При картографічному позиціонуванні виділяють два загальних способи відображення карт: геометричне і топологічне. На геометричній карті об'єкти відображаються у відповідності з їхніми абсолютними геометричними відношеннями. Це може бути сіткова карта або більш абстрактна лінійна чи полігональна карта. З іншого боку топологічне відображення більше базується на збереженні геометричних відношень між розпізнаними особливостями, ніж на їх абсолютне розташування в деякій системі відліку. На відміну від геометричних карт, топологічні карти можна будувати і підтримуватися без будь-якого положення робота. В результаті, цей підхід може використовуватися для побудови карт великих територій, так як всі зв'язки між вузлами швидше відносні, ніж абсолютні [3].

Висновок до розділу

Судячи з інформації поданої в цьому розділі, можна зробити висновок, що роботизовані системи почали своє існування досить давно, таким чином питання побудови навігації для керування системою постало відповідно під час побудови автономних мобільних роботів і до цих пір не існує тривіального рішення цієї проблеми.

В даний момент існує велика кількість способів для побудови навігації та використання її під час керування, але вони мають низку відмінностей, таких як вартість втілення, потреба в обчислювальних потужностях чи складності їх реалізації. За основу вирішення цього питання потрібно брати до уваги той факт, в якому місці, за яких умов, буде працювати система, спираючись на ці дані, обирається найбільш вигідніший спосіб для побудови навігації та керування нею.

					ІТ61020БАК.004 ПЗ	Арквш
Зм	Арк.	№ докum.	Підпис	Дата		16

2 ОГЛЯД ІСНУЮЧИХ РІШЕНЬ

Так як мобільні роботизовані пристрої зазнають широкого застосування в нашому повсякденному житті, існує безліч готових рішень та методик для розробки роботизованих систем, які використовують різними типи навігації під різні умови використання. Деякі готові рішення доступні на ринку розглянуті нижче.

2.1 Лінійна навігація

В даному способі навігації для побудови шляху руху використовуються маршрути у вигляді ліній, які наносяться на підлогу, ці лінії можуть бути виконані у вигляді яскравих білих ліній або ж можна нанести на підлогу магнітні лінії. Для реалізації такого способу компанія “RoboteQ” розробляє спеціальні датчики які застосовуються в лінійній навігації, платформа керування обладнується датчиком “RoboteQ MGS1600C”, який фіксує магнітне поле, (рис. 2.1).



Рисунок 2.1 – Датчик фіксування магнітних ліній [4]

					ІТ61020БАК.004 ПЗ	Архив
Зм	Арк.	№ докum.	Підпис	Дата		17

Порівняння декількох способів побудови маршрутів для лінійної навігації використовуючи датчики фіксування кольорових ліній, магнітних ліній та індукційних кабелів наведено в таблиці 2.1

Таблиця 2.1 – Порівняння способів побудови маршрутів

Назва	Легко наносити	Легко змінити	Безпечне	Стійке до бруду	Непомітне
Магнітна стрічка	Так	Так	Так	Так	Так
Індукційний кабель	Ні	Ні	Ні	Так	Так
Оптична стрічка	Так	Так	Так	Ні	Ні

З порівняння отримуємо, що найкращим способом є використання магнітних стрічок, які легко можна монтувати на підлозі, та легко побудувати маршрути переміщення для роботів, до того ж вони не використовують електроенергію та стійкі до забруднення, вони не помітні та їх можна розташувати під покриттям на підлозі, за умови якщо покриття підлоги зроблене з матеріалів, які не перешкоджають проходженню магнітних потоків.

Такий спосіб має досить значні недоліки, він вносить свої обмеження в форму роботизованої платформи, в різноманітність маршрутів, та неможливість уникати перешкод, які виникатимуть на шляху роботизованих пристроїв. Так як цей спосіб широко застосовується в індустріальній сфері де приміщення мають велику площу та працює невелика кількість людей то цей спосіб можна застосувати. Типи платформ які рекомендуються для застосування зображені на (рис. 2.2).

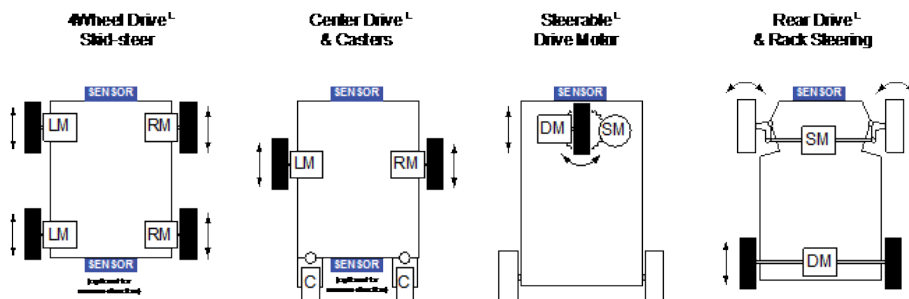


Рисунок 2.2 – Типи платформ для лінійної навігації [4]

Відповідно до кожної платформи виробник рекомендує розміщувати датчик магнітних ліній виключно як зображено на (рис. 2.2).

Керування курсом платформи дуже просте їй потрібно просто слідувати лінії та читувати команди в деяких місцях, датчик фіксування магнітних ліній повертає значення, які відповідають відстані до центру лінії. Ця інформація потім використовується для корегування курсу роботизованої платформи. Якщо платформа слідує паралельно лінії то значення з датчика дорівнюють 0. Для найкращої точності та мінімальної затримки, алгоритм керування може бути повністю замінений ПІД регулятором.

Керування прискоренням платформи залежить від середовища використання, але в даному випадку автоматизована платформа буде рухатись коли побачить стрічку повертатиме ліворуч або праворуч і зупиниться в точно визначеному місці відповідно до того як будуть нанесені маршрути переміщення в просторі. Платформа відновить свій рух через деякий час самостійно, або коли оператор натисне кнопку. Платформа постійно відслідковує наявність ліній на землі та відразу ж зупиниться, якщо лінія випадково закінчиться або буде перервана, такий спосіб більш безпечний ніж виконувати пошук самостійно.

Даний сенсор має вбудований алгоритм для розпізнавання та обробки розгалуження і сходження ліній. За замовчування контролер думає що лінії

					IT61020БАК.004 ПЗ	Автори
Зм.	Авк.	№ доквм.	Підпис	Дата		19

дві, зліва і справа. Коли платформа слідує одній лінії, датчик сприймає це як два паралельних маршрути. В той час як лінія починає розходитись, датчик сприймає це як розширення лінії з однієї сторони, та вимірює відстань між лівою та правою лінією, та відправляє команди в контролер який формує сигнал на мотори для здійснення зміни траєкторії, принцип роботи датчика та форми ліній зображений на (рис. 2.3).

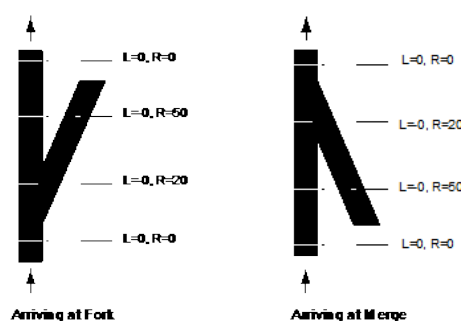


Рисунок 2.3 – Принцип роботи магнітного датчика [4]

Для надсилання команд платформі використовуються магнітні маркери, які є частиною магнітної стрічки з оберненою полярністю та розташовуються зліва і справа від центру основної лінії. Маркери це досить простий та ефективний спосіб для ідентифікації особливих точок вздовж маршруту.

В нашому випадку, використовуються маркери нанесені з лівого чи правого боку від основної лінії для подання платформі інформації, що попереду буде зміна напрямку, відповідно поворот наліво чи направо. Маркери, які розміщені по обидві сторони, сигналізують про місце де потрібно зробити зупинку (рис. 2.4).

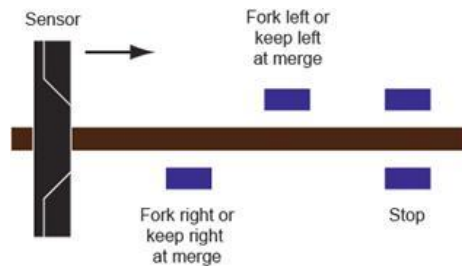


Рисунок 2.4 – Приклад розміщення командних маркерів [4]

Приклад простого тестового маршруту зі всіма доступними для системи командами зображено на (рис. 2.5).

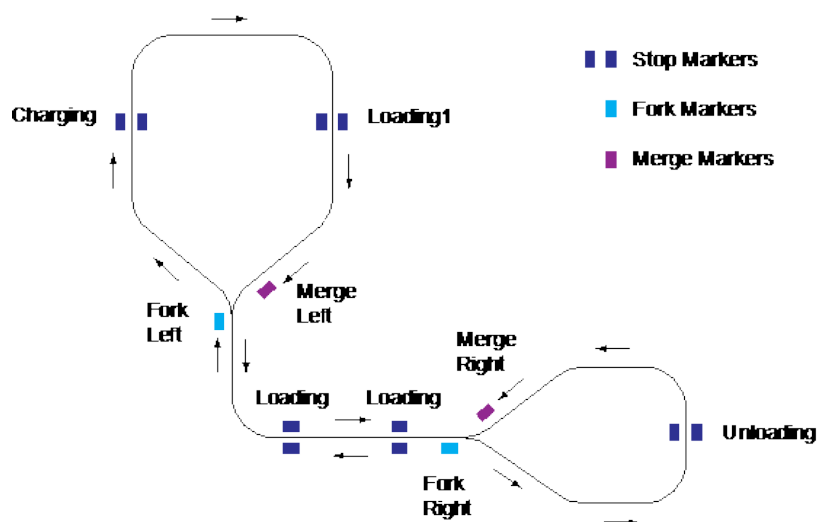


Рисунок 2.5 – Маршрут руху платформи [4]

2.1 Навігаційна система з використанням одометрії

Широкого застосування цей метод отримав в роботизованих системах, які переміщуються на відкритих площах та працюють повністю автономно або з мінімальним втручанням оператора. Таким чином у робота повна свобода дій

					IT61020БАК.004 ПЗ	Архив
Зм.	Арк.	№ док.м.	Підпис	Дата		21

він може самостійно будувати маршрут до заданої точки. Даний метод використовує підхід комбінування даних з декількох джерел, це дозволить мінімізувати кількість помилок при знаходженні власного місцеположення, але повністю не виключає їх.

Для цього використовується одометрія та електронний компас, дані з цих джерел комбінуються та поправляють один одного. Цей метод використовує два підходи для калібрування.

Перший підхід, це калібрування систематичної помилки одометрії для зменшення похибки в навігації та забезпечення точної оцінки положення робота, для подальшого застосування розширеного алгоритму фільтра Калмана. Наступний метод, калібрування електричного компаса з допомогою адаптивної нейронної нечіткої системи визначення, яка дає кут напрямлення мобільного робота для подальшого використання в алгоритмі фільтра Калмана. В кінці кінців, алгоритм корегування даних використовує фільтр Калмана, щоб поєднати дані з одометрії та електричного компаса, для отримання точної позиції та напрямку робота.

В додаток до цього, значення змінної k адаптивного розширеного алгоритму фільтра Калмана, яка відповідає коефіцієнту шуму, визначається нечітким алгоритмом.

Для демонстрації роботи даного способу використовується платформа мобільного робота, яка складається з електронного компаса, двох ультразвукових датчиків, які встановлені на кожній стороні для фіксування поточної траєкторії мобільної платформи. Також на роботі встановлено чотири мотори постійного струму та встановлені безпосередньо на осях моторів енкодери для вимірювання швидкості руху платформи (рис. 2.6).

					IT61020БАК.004 ПЗ	Архив
Зм	Арк.	№ докum.	Підпис	Дата		22

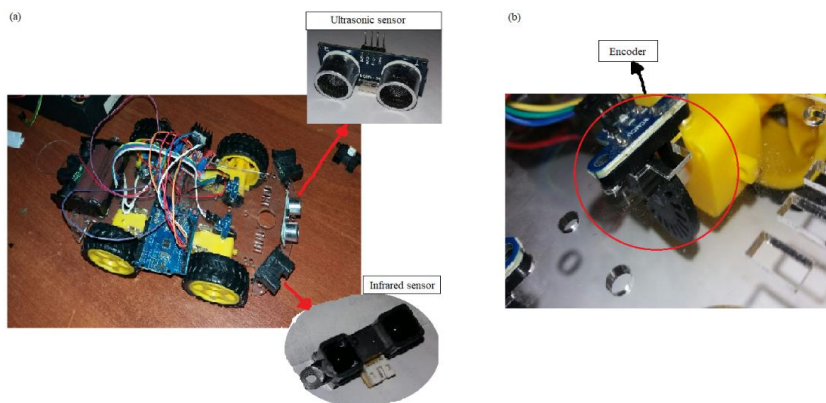


Рисунок 2.6 – Конструкція робота. а) Сенсори сприйняття.
б) Одометрична система

Як зображено на рисунку 2.6 два тягових колеса кріпляться на мотори постійного струму, і кожен енкодер встановлений безпосередньо на осі мотору, для найкращої обробки даних обертання колес. З отриманих імпульсів з енкодерів, можна вираховувати швидкість руху, відповідно до цього з допомогою формул можливе визначення положення та напрямлення робота.

Положення робота в конкретний проміжок часу позначається $x_i = [x_i \ y_i \ \theta_i]$, де x_i і y_i відповідні координати в Декартовій системі координат і θ_i - курс. На рисунку 2.7 зображено траєкторію переміщення двоколісної платформи з диференціальним приводом, на рисунку зображена ширина колісної бази, позначається B , зміна напрямку на кут $\delta\theta_i$ та пройдену дистанцію δD_i в конкретний відрізок часу i , маючи всі ці дані можна вирахувати положення робота використовуючи наступні формули.

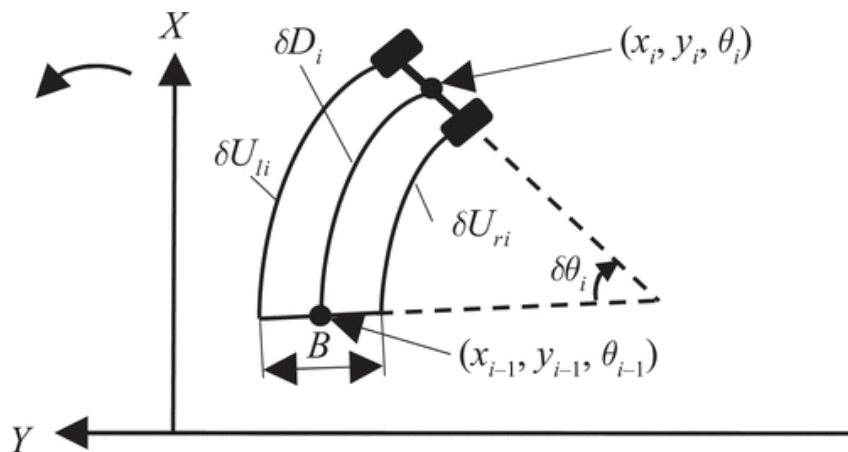


Рисунок 2.7 – Траєкторія переміщення робота [5]

$$\delta \theta_i = \frac{\delta U_{ri} - \delta U_{li}}{B}, \quad (2.1)$$

$$\delta D_i = \frac{\delta U_{ri} + \delta U_{li}}{2}, \quad (2.1) [6]$$

де δU_{ri} та δU_{li} відображають відстань пройдену роботом за певний відрізок часу під час переміщення лівим та правим колесами, та обчислюються за формулами:

$$\delta U_{ri} = \frac{\delta N_{ri} D_r \pi}{n}, \quad (2.3)$$

$$\delta U_{li} = \frac{\delta N_{li} D_l \pi}{n}, \quad (2.3) [6]$$

де δN_{ri} та δN_{li} - кількість імпульсів енкодера лівого та правого колеса відповідно. D_r та D_l - діаметр лівого та правого колеса (в міліметрах). n - кількість імпульсів енкодера на один оберт.

Відповідно до рисунку 2.7, переміщення робота в i -й відрізок часу можна вирахувати за формулами.

					ІТ61020БАК.004 ПЗ	Арквш
Зм	Арк.	№ доквм.	Підпис	Дата		24

$$\delta x_i = \frac{\delta D_l}{\delta \theta_l} [\sin(\theta_{i-1} + \delta \theta_i) - \sin(\theta_{i-1})], \quad (2.4) [6]$$

$$\delta y_i = \frac{\delta D_l}{\delta \theta_l} [\cos(\theta_{i-1}) - \cos(\theta_{i-1} + \delta \theta_i)], \quad (2.5) [6]$$

Далі нам потрібен електричний компас, це датчик, який вказує на азимут вимірюючи магнітне поле землі. В даному випадку використовується модель компаса DCM230B.

Шляхом заводського калібрування, програмного калібрування та калібрування похибки на магнітні перешкоди, вдалося довести розбіжність значень азимуту до $\pm 1^\circ$, точність азимуту 1° , в навігації допускається кут нахилу в $\pm 5^\circ$. Цей компас здатен працювати в середовищах з температурним діапазоном від -20° до 70° . Але на одному компасам система працювати не зможе тому далі розглянемо роботу одометрії.

Існує три відомих систематичних помилок які впливають на точність одометричних даних: різні діаметри коліс, неправильно визначена ширина колісної бази, різниця між середнім значенням фактичних діаметрів коліс з номінальним значенням діаметрів. Далі розглянемо метод для виправлення цих трьох домінуючих помилок.

Дві основні систематичні помилки, які впливають на прямолінійний рух робота. Це різний діаметр коліс буде вносити коректив в напрямок робота замість руху по прямій. Та неправильно визначений діаметр коліс, в наслідок цього дистанція робота отримана з енкодерів буде відрізнятись від реального пройденого шляху. Як показано на рисунку 2.8, вважаємо що ліве колесо має більший діаметр ніж праве для того щоб розглянути випадок коли робот має джерело помилки в якості різниці діаметрів коліс або ж різницю в потужності моторів, тому траєкторія робота буде мати вплив цих факторів.

В даний момент у нас ідеальні умови і відкинемо, ще одне джерело похибки, як прослизання коліс. Тому позначимо номінальний шлях як L і

					ІТ61020БАК.004 ПЗ	Архив
Зм	Арк.	№ докум.	Підпис	Дата		25

справжній шлях як L_{ca} . Далі позначимо відхилення від прямої траєкторії, як C та кут, який відповідає цьому криволінійному переміщенню як φ .

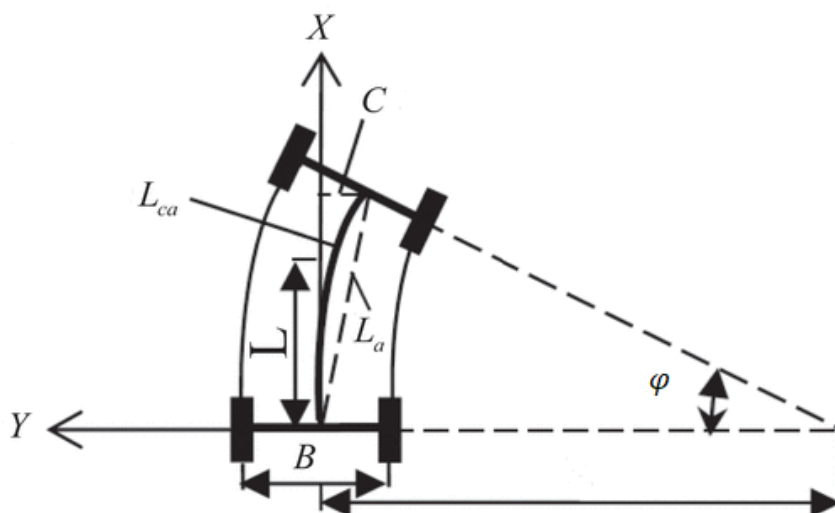


Рисунок 2.8 – Рух робота по прямій лінії

Позначмо криволінійне переміщення лівого та правого колеса як L_l^l та L_r^l відповідно, тоді з допомогою наступних формул можна визначити пройдений шлях кожного колеса

$$L_l^l = \frac{N_l^l D_l^a \pi}{n}, \quad (2.6) [6]$$

$$L_r^l = \frac{N_r^l D_r^a \pi}{n}, \quad (2.7) [6]$$

де N_l^l і N_r^l кількість імпульсів енкодера на лівому та правому колесі.
 D_l^a та D_r^a діаметр лівого та правого колеса, відповідно в міліметрах.

					ІТ61020БАК.004 ПЗ	Архив
Зм.	Арк.	№ док.	Підпис	Дата		26

n – кількість імпульсів енкодера на один оберт. На рисунку 2.8 ми маємо розмір колісної бази B , шлях пройдений роботом L_{ca} та кут φ , можна обчислити за формулами.

$$L_{ca} = \frac{L_l^l + L_r^l}{2} = \frac{\pi(N_l^l D_l^a + N_r^l D_r^a)}{2n}, \quad (2.8) [6]$$

$$\varphi = \frac{L_r^l - L_l^l}{B} = \frac{\pi(N_r^l D_r^a - N_l^l D_l^a)}{nB}, \quad (2.9)$$

Основні формули для розуміння принципу роботи роботизованої системи з диференціальним приводом розглянуто.

Висновок до розділу

Під час дослідження існуючих рішень, були розглянуті два найпопулярніших способи побудови та використання навігації в мобільних роботі, кожен з цих способів має свої характерні відмінності, які надають їм переваги під час використання в певних умовах.

Навігація по лініям – просто реалізовується, але має свої обмеження у вигляді свободи перебудови маршрутів роботом власноруч.

Навігація побудована на одометрії – дає більший простір для дій, таким чином робот вже може з допомогою відповідних алгоритмів, аналізуючи ці дані будувати найоптимальніший маршрут, щоб дістатись з пункту А в пункт Б. Але спосіб використовувати тільки одну одометрію має недоліки, у вигляді накопичення помилки під час руху із-за похибки в датчиках, та вплив умов використання на систему.

					ІТ61020БАК.004 ПЗ	Архив
Зм	Арк.	№ докум.	Підпис	Дата		27

3 РОЗРОБЛЕННЯ СИСТЕМИ КЕРУВАННЯ МОБІЛЬНОЇ ПЛАТФОРМИ

3.1 Дослідження інструментів для реалізації проєкту

Під час дослідження теми проєкту було розглянуто багато методик розробки подібних систем для керування роботизованою платформою та було виявлено ряд недоліків кожного способу реалізації для керування мобільними роботизованими системами.

Так як мобільні роботизовані системи швидко набирають популярність, існує багато способів розробки таких систем, деякі команди для розробки використовують просто мікроконтролери і пишуть всю логіку на них, такий спосіб складний та часозатратний. Такий спосіб має свої недоліки такі як, складність переносу системи на інші пристрої та потреба маніпулювати багатьма сутностями під час розробки.

Проаналізувавши різні способи реалізації, команда розробників з Willow Garage запропонувала свій гнучкий інструмент для створення роботизованих систем, ROS – robot operating system, перша версія фреймворку вийшла в 2007 році та розвивається по цей час і завойовує все більшу прихильність розробників.

Операційна система для роботів – це програмне забезпечення для розробки роботизованих систем, тобто колекція фреймворків для розробки програмного забезпечення роботів. Хоча ROS сама по собі не являється операційною системою, вона надає сервіси розроблені для неоднорідних комп'ютерних кластерів, таких як апаратне абстрагування, керування низькорівневими пристроями, також має реалізовані часто вживані функціональності, обмін повідомленнями між процесами та керування пакетами. Всі процеси які працюють в ROS, представлені в вигляді графа, де обробка даних відбувається у вузлах, які можуть приймати, відправляти та проводити маніпуляції з даними, отриманими з датчиків, керувати станом системи та обробляти інші повідомлення.

					IT61020БАК.004 ПЗ	Арквш 28
Зм	Арк.	№ докум.	Підпис	Дата		

Незважаючи на важливість швидкодії та низьку затримку під час керування роботами, сама ROS не є операційною системою реального часу (RTOS). Однак її можна інтегрувати з кодом написаним для мікроконтролера, який працює в реальному часі. Хоча на даний момент часу, відсутність підтримки систем у режимі реального часу вирішено версією ROS 2. Також плюс цього інструменту в простоті переносу проєктів на інші платформи, це забезпечується тим, що ROS підтримується на всіх Unix-подібних системах, таким чином ми можемо використовувати ROS на будь-якому пристрої де встановлена Linux система.

Також для спрощення використання під окремі задачі у фреймворка ROS з'явилося відгалуження у вигляді вузьконаправленого фреймворка, такого як ROS-Industrial, який застосовується для розробки програмного забезпечення виключно для індустріальних маніпуляторів.

Цей інструмент має три концептуальних рівні: рівень файлової системи, рівень обчислень та рівень взаємодії [8]. Ці рівні та концепції наведені далі. Концепції рівня файлової системи в основному охоплюють ресурси ROS, з якими ви працюєте під час розробки:

- Packages: Пакети - це основні блоки, які формують програмне забезпечення в ROS, пакети це найменші частинки в інфраструктурі роса всі реалізовані модулі поміщаються в пакти та можуть бути задіяні в інших системах;
- Packages Manifests: Маніфести (package.xml) містять метадані про пакет, опис, версію, автора;
- Message (msg): Всі описи повідомлень зберігаються в окремій папці в файлах типу .msg;
- Service (srv): Описи сервісів, що зберігаються в папці srv типу .srv, визначають структури даних про запити та відповіді для сервісів в ROS.

					IT61020БАК.004 ПЗ	Архив
Зм	Арк.	№ докум.	Підпис	Дата		29

3.1.1 Архітектура ROS

Архітектура системи утворює однорангову мережу процесів ROS, в якій виконується обробка даних разом. Основні частини з яких складається система є вузли, майстер вузол, сервер параметрів, повідомлення, сервіси, топіки та пакети, всі вони взаємодіють один з одним та обмінюються даними.

- Nodes: Вузли - це процеси, в яких виконуються обчислення. ROS розроблений таким чином, щоб бути модульним у дрібнозернистих масштабах; система керування роботом зазвичай містить велику кількість вузлів;

- Master: Master вузол ROS забезпечує реєстрацію імен та забезпечує комунікацію всіх вузлів між собою;

- Parameter Server: Сервер параметрів - це спільний багатозмінний словник, який доступний через мережеві API;

- Messages: Вузли спілкуються один з одним, передаючи повідомлення. Повідомлення - це просто структура даних, що включає поля з даними;

- Topics: Повідомлення передаються через транспортну систему з семантикою публікації/підписки;

- Services: Модель передачі/підписки – це дуже гнучка парадигма комунікації, проте її багатосторонній, односторонній транспорт не підходить для взаємодії запит/відповідь, які часто потрібні в розподіленій системі.

Майстер ROS виступає в якості сервісу імен в графі обчислень ROS. Він зберігає інформацію про реєстрацію тем та сервісів для вузлів ROS. Вузли спілкуються з Майстром, щоб повідомити інформацію про їх реєстрацію. Оскільки ці вузли спілкуються з Master вузлом, вони можуть отримувати інформацію про інші зареєстровані вузли та здійснювати з'єднання у відповідних випадках. Master вузол також буде робити зворотні виклики цим вузлам, коли ця інформація про реєстрацію змінюється, що дозволяє вузлам

					IT61020БАК.004 ПЗ	Аркvш
Зм	Арк.	№ докum.	Підпис	Дата		30

динамічно створювати з'єднання під час запуску нових вузлів. Вузли підключаються безпосередньо до інших вузлів.

Master працює по типу DNS-сервера він надає інформацію для пошуку вузлів в мережі яка створюється під час роботи головного ядра ROS. Вузли, які підписуються на тему, вимагають з'єднання до вузлів, які публікують дані до цієї теми і встановлюють з'єднання через узгоджений протокол з'єднання. Найпоширеніший протокол, що використовується в системі, називається TCPROS, який використовує стандартні сокети TCP/IP. Ця архітектура дозволяє здійснювати нерозв'язне функціонування, де назви є основним засобом, за допомогою якого можна будувати більші та складні системи. Імена відіграють дуже важливу роль у ROS: вузли, теми, послуги та параметри мають усі імена.

Кожна бібліотека ROS підтримує перезапис імен тем та командних рядків, що означає, що створена програма може бути переконфігурована під час виконання для роботи в іншій топології графа обчислень, така можливість дає гнучкий підхід до розробки модулів, які можна не переписувати по новій, а просто переконфігурувати для роботи в іншій топології. Концепція роботи фреймворка (рис. 3.1) головний Майстер вузол з'єднує між собою всі вузли низького рівня, які утворюють між собою мережу та можуть обмінюватись повідомленнями різних типів даних.

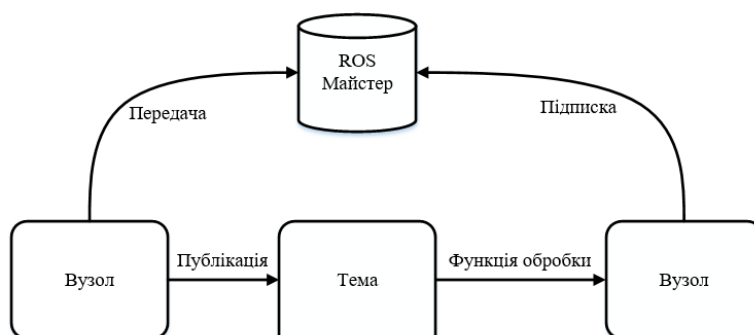


Рисунок 3.1 – Концепція роботи ROS

До того ж ROS має досить хороший інструментарій для розробки та тестування програмного забезпечення Rviz та Rqt.

Rviz – графічний інструмент для відображення поточного стану системи в реальному часі.

Rqt – являє собою програмний фреймворк ROS, який реалізує різні інструменти графічного інтерфейсу у вигляді плагінів. Можна запустити всі існуючі інструменти графічного інтерфейсу як набір панелей для керування різними частинами системи. Інструменти також можуть працювати в традиційному автономному режимі, але rqt спрощує керування одразу всіма різними вікнами на екрані.

Також можна використовувати симуляцію під час розробки програмного забезпечення, для цього запропонований симулятор Gazebo. Він дозволяє розробити систему та провести максимальну кількість можливих тестів не будуючи реальну модель робота, це заощаджує час та кошти під час розробки, на рисунку 3.2 зображено віртуальну модель робота в симуляторі Gazebo.

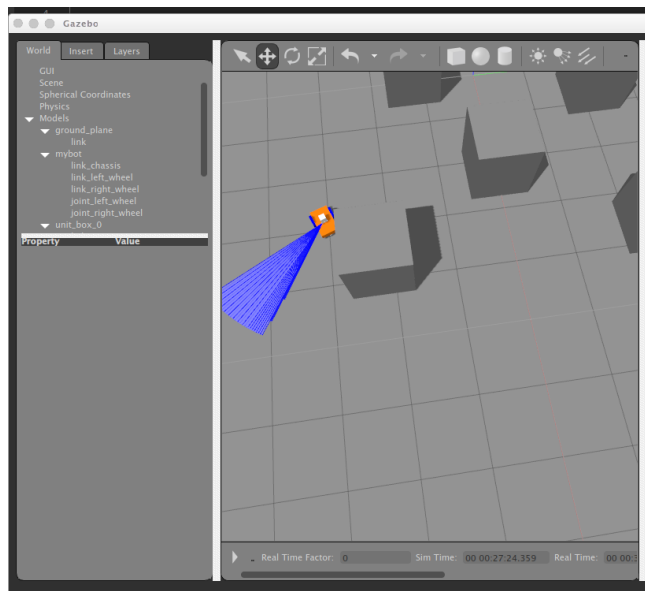


Рисунок 3.2 – Gazebo симулятор з роботом

Так як тема проекту пов'язана з роботизованими системами, для реалізації цієї системи керування, фреймворк ROS є найкращим інструментом.

Можливість реалізації проекту в симуляторі дає велику кількість переваг та заощадити час на експерименти.

Для переносу готової системи нам знадобиться всього пристрій на якому встановлено Linux та який має можливість працювати з GPIO портами або ж можна реалізувати взаємодію з GPIO використовуючи звичайний мікроконтролер та підключити його до комп'ютера через USB кабель та передавати дані.

3.2 Хід виконання проекту

3.2.1 Розробка діаграми активності

На кресленнику IT61.020БАК.004 Д1 представлена діаграма діяльності системи. На даній діаграмі зображено роботу в основному робочому режимі, коли ми можемо здійснювати налаштування та налагодження системи.

При запуску системи, вона починає свою роботу із завантаження головних модулів, які створюють мережу в якій будуть працювати подальші підпрограми. Коли всі основні модулі завантажені, система проводить самодіагностику на працездатність та видає результат своєї готовності.

Якщо результат не задовільний система видає сповіщення про несправності та просить втрутитися оператора. Коли результат задовільний система завантажується та переходить в стан очікування, в даний момент можна виконати першочергові налаштування системи такі як, додати інших роботів для створення мережі між ними, в якій вони зможуть обмінюватись даними один з одним, задати обмеження по площі переміщення робота, перевести в режим ручного керування або автономного під час якого робот самостійно виконує всі дії.

					IT61020БАК.004 ПЗ	Аркvш
						33
Зм	Арк.	№ докum.	Підпис	Дата		

Після проведення налаштувань, систему можна перевести в робочий режим, вона здійснить з'єднання з іншими роботами в системі та повністю готова до використання.

В робочому режимі доступні основні функції, які відповідають за переміщення робота, видачі йому завдань, збір інформації про навколишнє середовище, сканування карти, перегляд результатів роботи системи, шляхи переміщення або відображення положення та стану системи в реальному часі.

Під час сканування та збору інформації про навколишнє середовище можна обрати два способи роботи системи в ручному режимі та автономному.

В ручному режимі всі дії по переміщенню робота покладені на руки оператора, цей спосіб має свої переваги в тому, що людина може швидше оцінити, шлях оптимального переміщення для збору інформації середовища, а під час автономного керування робот сам буде переміщуватись використовуючи базовий алгоритм дослідження середовища.

3.2.2 Розробка алгоритму побудови маршруту

На кресленнику IT61.020БАК.004 Д2 представлена блок-схема алгоритму побудови маршруту, даний алгоритм може працювати коли система знаходиться в робочому режимі та можна давати їй завдання на виконання.

Користувач використовуючи графічний інструмент або API модуля навігації для задання положення з коду. Передає в метод координати точки в яку робот повинен переміститись. Після чого відбувається перше сканування навколишнього середовища, яке знаходиться в радіусі двох метрів від самого робота. Потім запускається алгоритм оцінки положення, тобто відбувається зіставлення фактичного положення робота з приблизним положенням на мапі. Якщо положення знаходиться в нормі, робот знаходиться безпосередньо в точці яка співпадає з точкою в реальному середовищі, результат оцінки може приймати значення в межах від нуля до одного. Нуль – означає робот

						Адквш
Зм	Арк.	№ доквм.	Підпис	Дата	IT61020БАК.004 ПЗ	34

знаходиться в неправильному положенні, тому потрібно провести калібрування системи та зайняти правильне положення. Один означає, що всі значення співпали та робот правильно виконав позиціонування в середовищі і він повністю готовий до роботи по переміщенню в просторі.

Якщо оцінка більша 0.5 відбувається визначення приблизної відстані до зазначеної точки на карті. Потім виконується побудова можливих маршрутів використовуючи алгоритм Дейкстри, набір маршрутів переходить для обробки та визначення найкоротшого відносно початкової відстані від поточного положення робота до заданої точки.

Після чого найкоротший маршрут передається на виконання в модуль переміщення де вже робот використовує локальне позиціонування в радіусі двох метрів і якщо на шляху з'являються перешкоди робот самостійно враховує цей факт та перебудовує маршрут в цьому радіусі для уникнення зіткнення з перешкодою.

3.2.3 Розробка діаграми прецедентів системи

Діаграма доступного функціоналу системи, представлена на кресленику ІТ61.020БАК.004 Д4, детальний опис кожної доступної функції наведено нижче у таблицях 3.1 – 3.9.

Таблиця 3.1 – Варіант використання «Задання точки»

Назва	Задання точки на карті
Опис	В графічному інструменті вибираємо інструмент для задання позиції на карті до якої повинен рухатись робот та розміщуємо її на карті.
Учасники	Користувач
Мета	Визначення положення де має бути платформа в результаті руху.

						ІТ61020БАК.004 ПЗ	Архив
Зм	Арк.	№ докum.	Підпис	Дата			35

Передумови	1) В наявності є побудована карта приміщення в якому відбувається робота. 2) Система знаходиться в робочому режимі 3) Запущений процес побудови шляху. 4) Користувач з допомогою графічного чи вручну задає точку на карті до якої потрібно переміститися.
Постумови	Робот здійснив переміщення по карті в задану точку та завершив виконання маневру.
Основний сценарій	1) Перейти в режим задання маршруту переміщення. 2) Користувач натискає кнопку «Set goal». 3) Користувач визначає місце на карті та ставить точку. 4) Робот обробляє запит, будує маршрут повертає код виконання операції та починає переміщення.

Таблиця 3.2 – Варіант використання «Запуск сканування»

Назва	Запустити автономного сканування середовища
Опис	Буде виконувати сканування середовища для відображення карти автономно.
Учасники	Користувач
Мета	Створити відображення карти середовища для подальших переміщень робота на ній.
Передумови	1) Система знаходиться в робочому режимі. 2) Дані одометрії стерті в нуль. 3) Задані початкові налаштування для сканування середовища. 4) Обраний режим керування роботом під час сканування. 5) Запущена команда сканування.

						ІТ61020БАК.004 ПЗ	Архив
Зм.	Арк.	№ докум.	Підпис	Дата			36

Постумови	Робот здійснив переміщення по карті під час яких сканував середовище та відображав його на карті. Готова карта середовища.
Основний сценарій	1) Перейти в режим сканування. 2) Скинути попередні дані. 3) Обрати режим керування роботом під час переміщення. 4) Користувач запускає команду «scan». 5) Робот переміщується по середовищу та скануючи робить відображення на карті.

Таблиця 3.3 – Варіант використання «Створення набору точок»

Назва	Створити набір точок відвідування
Опис	Користувач може створити маршрут який складається з багатьох точок.
Учасники	Користувач
Мета	Створити багато точок на карті для відвідування системою.
Передумови	1) В наявності є побудована карта приміщення в якому відбувається робота. 2) Система знаходиться в робочому режимі 3) Запущений процес побудови шляху. 4) Користувач з допомогою графічного чи вручну задає точки на карті до яких потрібно переміститися роботу.
Постумови	Робот здійснив переміщення по карті в заданому маршруті, дістався останньої точки та завершив виконання маневру.
Основний сценарій	1) Перейти в режим задання маршруту переміщення. 2) Користувач натискає кнопку «Set many goals». 3) Користувач визначає місця на карті та позначає точки. 4) Робот обробляє запит, будує маршрут повертає код виконання операції та починає переміщення.

						ІТ61020БАК.004 ПЗ	Архів
Зм.	Арк.	№ докум.	Підпис	Дата			37

Таблиця 3.4 – Варіант використання «Збереження результатів»

Назва	Зберегти поточні результати
Опис	Користувач може зафіксувати поточний стан системи для налагодження.
Учасники	Користувач
Мета	Зберегти інформацію про поточний стан системи для налагодження.
Передумови	1) Система знаходиться в робочому режимі 2) Запущений процес діагностики. 3) В наявності зібрана інформації про поточний стан системи. 4) Користувач запустив команду збереження інформації.
Постумови	В файлі логів збережено поточну інформацію про стан системи.
Основний сценарій	1) Перейти в режим очікування. 2) Запустити команду «Save state». 3) Користувач проводить налаштування, яку інформацію зберегти. 4) Інформація записується до «bag» файлу.

Таблиця 3.5 – Варіант використання «Перегляд результатів переміщення»

Назва	Переглянути результати переміщення робота
Опис	Користувач може відобразити хронологію переміщення робота в просторі.
Учасники	Користувач
Мета	Створити файл який містить інформацію про переміщення робота з поточний сеанс.
Передумови	1) В наявності є побудована карта приміщення в якому відбувається робота.

						ІТ61020БАК.004 ПЗ	Арквш
Зм	Арк.	№ доквм.	Підпис	Дата			38

	2) Система знаходиться в режимі очікування. 3) Запущений процес збирання даних за останній сеанс..
Постумови	Після здійснення переміщень робота в поточному сеансі отримано відображення шляхів переміщення в просторі.
Основний сценарій	1) Перейти в режим задання маршруту переміщення. 2) Користувач натискає кнопку «Set many goals». 3) Користувач визначає місця на карті та позначає точки. 4) Робот обробляє запит, будує маршрут повертає код виконання операції та починає переміщення.

Таблиця 3.6 – Варіант використання «Перегляд останнього відвіданого місця»

Назва	Переглянути останнє відвідане місце роботом
Опис	Користувач може переглянути останнє місцезнаходження робота за поточний сеанс.
Учасники	Користувач
Мета	Відобразити успішність досягнення роботом заданої точки на карті середовища та простоти налагодження.
Передумови	1) В наявності є побудована карта приміщення в якому відбувається робота. 2) Система знаходиться в режимі очікування. 3) Запущений процес збирання даних за останній сеанс.
Постумови	Після виконання команди відображено останнє місцезнаходження робота та стан на той момент часу.
Основний сценарій	1) Перейти в режим очікування. 2) Користувач запускає команду «Show_last_point». 3) Система проводить діагностику системи та збирає дані за останній сеанс. 4) Дані записуються в файл та можуть бути відображені у вигляді рисунку.

						ІТ61020БАК.004 ПЗ	Архив
Зм.	Арк.	№ док.	Підпис	Дата			39

Таблиця 3.7 – Варіант використання «Повернутися в стартову точку»

Назва	Повернутися в стартову точку
Опис	Користувач може дати команду роботу для повернення в стартову точку.
Учасники	Користувач
Мета	Повернути робота в стартову точку по тому ж маршруту.
Передумови	1) В наявності є побудована карта приміщення в якому відбувається робота. 2) Система знаходиться в режимі очікування. 3) Робот дістався заданої точки.
Постумови	Після виконання команди положення робота повинно співпадати зі стартовим положенням.
Основний сценарій	1) Перейти в режим задання точки. 2) Користувач натискає кнопку «Move back». 3) Система подає на обробку останній маршрут, так як він був задіяний останнім. 4) Робот виконує переміщення до стартової точки на карті.

Таблиця 3.8 – Варіант використання «Перейти на ручне керування»

Назва	Перейти на ручне керування
Опис	Користувач може перевести систему в ручне керування та здійснювати переміщення робота з використанням пульта або клавіатури комп'ютеру.
Учасники	Користувач
Мета	Перевести систему в ручне керування, для керування оператором.
Передумови	1) Система переведена в режим очікування. 2) В систему підключено пульт керування. 3) Запущено команду обробки команд від оператора.

					IT61020БАК.004 ПЗ	Архив
Зм.	Арк.	№ докum.	Підпис	Дата		40

Постумови	Після виконання команди керувати роботом можна з допомогою пульта або клавіатури.
Основний сценарій	1) Перейти в режим очікування. 2) Користувач запускає команду «teleop_cmd». 3) Система переходить в ручне керування. 4) Керування роботом здійснюється з допомогою пульта.

Таблиця 3.9 – Варіант використання «Запуск ручного сканування»

Назва	Запуск ручного сканування середовища
Опис	Буде виконувати сканування середовища для відображення карти в ручному режимі.
Учасники	Користувач
Мета	Створити відображення карти середовища для подальших переміщень робота на ній.
Передумови	1) Система знаходиться в робочому режимі. 2) Дані одометрії стерті в нуль. 3) Задані початкові налаштування для сканування середовища. 4) Обраний режим ручного керування роботом під час сканування. 5) Запущена команда сканування.
Постумови	Робот здійснив переміщення по карті під час яких сканував середовище та відображав його на карті.
Основний сценарій	1) Перейти на ручне керування. 2) Перейти в режим сканування. 3) Скинути попередні дані. 4) Користувач запускає команду «scan». 5) Робот переміщується по середовищу та скануючи робить відображення на карті.

					ІТ61020БАК.004 ПЗ	Архив
Зм.	Арк.	№ доквм.	Підпис	Дата		41

3.2.4 Побудова діаграми розгортання

На кресленнику IT61.020БАК.004 ДЗ представлено діаграму розгортання системи, дана діаграма відображає два хоста, головний комп'ютер та комп'ютер який встановлений на роботі, також модулі з яких складаються системи. На головному комп'ютері встановлений головний модуль, який створює мережу в якій працюють дочірні модулі. Керування роботом можна здійснювати з головного комп'ютеру використовуючи команди терміналу, перед тим утворивши з'єднання безпосередньо з роботом або ж використовувати графічні інструменти, які мають в своєму арсеналі набір команд, які представлені у вигляді графічного інтерфейсу для спрощення керування звичайному користувачу.

На самому роботі реалізовано набір модулів таких як модуль обробки команд користувача, модуль побудови маршруту, навігаційний модуль та модуль збереження даних та логів.

В модулі обробки команд реалізовано безпосередньо обробку доступних користувачеві команд та перетворення їх в сигнали на мотори робота.

Модуль побудови шляху реалізовує в собі алгоритм пошуку найкоротшого шляху, для переміщення до точки на карті, яку задав користувач.

Навігаційний модуль відповідає за збір інформації довколишнього середовища, побудови карти та позиціонування системи на побудованій мапі, в даній системі навігація є основним компонентом, так як система може пересуватись в автономному режимі приймаючи самостійно рішення про перебудову маршруту для уникнення зіткнень з перешкодами або раптовою зміною ситуації навколо.

Інтерфейс даних використовується для обміну даними в середині системи між внутрішніми модулями.

					IT61020БАК.004 ПЗ	Архив
Зм	Арк.	№ доквм.	Підпис	Дата		42

3.2.5 Реалізація системи керування

Перед початком реалізації потрібно виділити основні аспекти з яких повинна складатися система, відповідно до цієї інформації визначити, які компоненти потрібно реалізувати та розробити архітектуру самої моделі роботизованої системи та програмного забезпечення.

Для реалізації проєкту обраний тип роботизованої системи на колісній базі, керування системою буде відбуватися віддалено оператором або автономно з заданням точки на карті до якої повинен приїхати робот.

Для даної роботизованої системи потрібні такі модулі:

- Зчитування команд від користувача;
- Розпізнавання команд в сигнали для руху робота;
- Модуль навігації;
- Модуль Одометрії.

Так як ROS має велику кількість допоміжних інструментів деякі модулі вже реалізовані та можна вбудовувати їх в систему. Тому модулі зчитування команд користувача, відображення навігації, ми можемо взяти з бібліотеки ROS, всі інші модулі потрібно реалізувати власноруч.

Компоненти системи які потребують реалізації визначені, наступним кроком, потрібно розробити опис роботизованої платформи, включити всі потрібні сенсори для коректного функціонування системи, потім створити віртуальну модель робота в симуляторі Gazebo. Для цього з допомогою мови розмітки в URDF – файлі, повністю описати всі частини робота, вказати точки з'єднання між всіма частинами та задати їхні фізичні характеристики для найкращих даних, які максимально приближені до реальних. Готова модель роботизованої платформи зі всіма потрібними датчиками зображена на (рис. 3.7).

					ІТ61020БАК.004 ПЗ	Аркуш
Зм	Арк.	№ докum.	Підпис	Дата		43

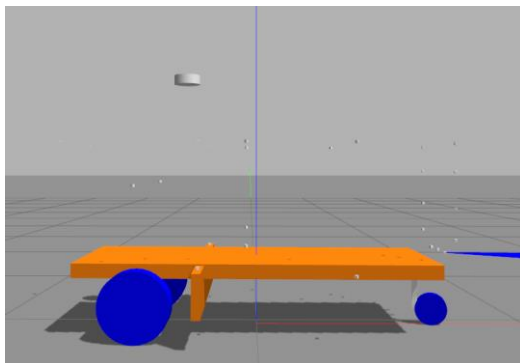


Рисунок 3.7 – Модель роботизованої платформи

Модель описується в URDF- файлі з використанням мови розмітки XML, з допомогою спеціально створених тегів, формується фрейми системи та з'єднуються між собою, кожен фрейм має свої характеристики.

Joint – спеціальний тег в мові розмітки XML, який використовується для з'єднання елементів між собою та виконує опис кінематики та динаміки з'єднання, визначає межі безпечного руху з'єднань.

За цим принципом побудована модель робота яка зображена на рисунку 3.4, кожна частинка має свої характеристики такі як, форма, вага, центр маси, в свою чергу з'єднувальні елементи також мають налаштування, вони містять інформацію про частинки які потрібно поєднати, в якому місці вони повинні бути поєднані та тип з'єднання, фіксований, обертальний чи лімітований обертальний.

Після того як модель робота готова, можна переходити до реалізації програмної частини. Спочатку потрібно реалізувати модуль обробки команд та перетворення їх в сигнали на мотори платформи, для отримання даних з енкодерів та формувати одометрію, яка допоможе корегувати його рух.

Для того щоб зрозуміти яким чином потрібно обробляти дані, для отримання правильних значень, потрібно розглянути, як система орієнтується в просторі та як буде визначати власне положення і яким чином вираховує

					IT61020БАК.004 ПЗ	Архив
Зм	Арк.	№ докum.	Підпис	Дата		44

свою швидкість та координати переміщення. Для цього нам потрібно розглянути кінематику руху робота.

Отримання моделі для руху всього робота - це процес від основного вузла до кінцевих подібне до графу дереву. Кожне колесо сприяє рухові роботи і в той же час накладає обмеження на рух. Колеса пов'язані між собою на платформі шасі робота, а тому обмеження їхнього переміщення поєднуються, утворюючи обмеження на рух всього робота. Але рухи та обмеження кожного колеса повинні бути виражені щодо чіткої та послідовної системи відліку. Це особливо важливий аспект в мобільній робототехніці, через її можливість автономного переміщення, потрібне чітке відображення між глобальними та локальними системами координат. Почнемо з того, щоб визначити ці системи формально, потім використовуючи отримані данні для обчислення кінематики окремих коліс і всього робота [10].

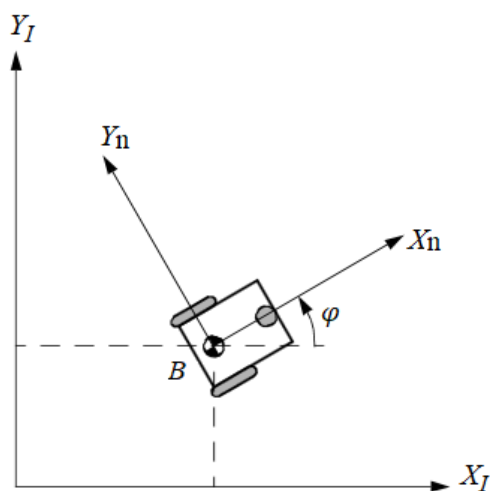


Рисунок 3.8 - Глобальна система координат та локальна система координат робота

Зм.	Арк.	№ докum.	Підпис	Дата

IT61020БАК.004 ПЗ

Аркvш

45

Мобільний робот має шість ступеней свободи. Здійснює переміщення по трьом осям координат x , y , z , які дозволяють визначити положення об'єкта в три вимірному просторі. Наступні три степені свободи стосуються орієнтації робота в просторі. Такі значення як нахил на сторону, нахил вперед назад, та відхилення від заданої траєкторії. Платформа з диференціальним приводом здійснює переміщення по двовірному просторі, і в даний момент положення робота можна описати з допомогою двох глобальних координат X , Y , розташованих в горизонтальній площині. Та курс робота позначається φ . Цих даних вистачає, щоб описати положення робота з диференціальним приводом (рис. 3.9).

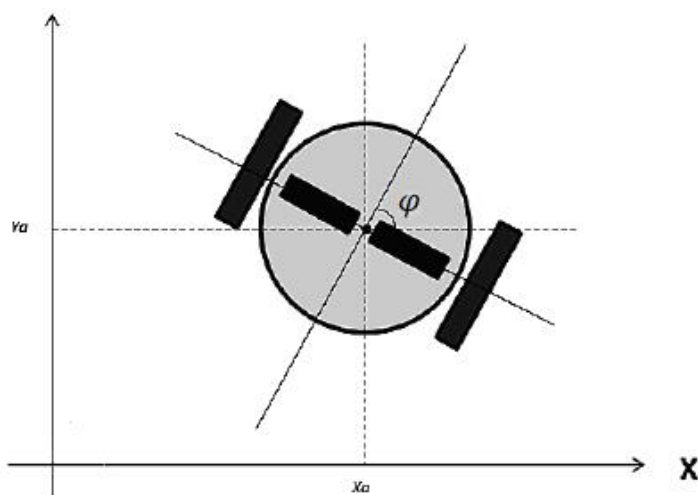


Рисунок 3.9 - Положення робота x , y , тета в глобальній системі координат

Диференціальний привод робота заснований на різниці швидкостей правого та лівого ведучого колеса [11]. При зміні швидкостей одного з коліс, робот буде змінювати напрямок в ту сторону, з якої колесо крутиться повільніше. Далі нам потрібно визначити розміри коліс та бази платформи,

					ІТ61020БАК.004 ПЗ	Аркvш
Зм	Арк.	№ докum.	Підпис	Дата		46

відповідно до рівнянь кінематики робота, за один оберт колесо проходить відстань рівну $2\pi r$, де r – радіус колеса. Коли робот здійснює поворот навколо точки, яка знаходиться на осі одного з коліс. Ця точка знаходиться за межами робота та називається миттєвим центром кривої. На (рис. 3.10) зображено конфігурацію тягових коліс з диференціальним приводом по відношенню до точки навколо якої здійснюється поворот.

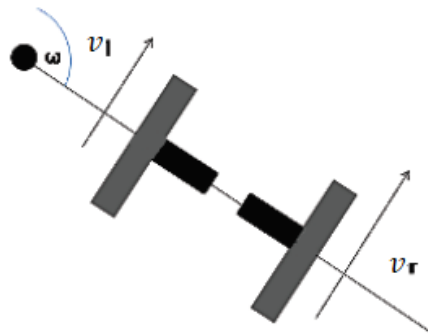


Рисунок 3.10 - Конфігурація коліс робота з диференціальним приводом

Головне поняття, без якого не вийде отримати рівняння кінематичного приводу - це кутова швидкість. Коли робот виконує поворот, тягові колеса рухаються по колу, центр якого співпадає з центром кривої. Швидкість колеса описується рівнянням $v = \frac{2\pi r}{T}$, T – час, затрачений на проходження колесом відстані, рівній довжині повного кола з центром, який співпадає з центром кривої. Відповідно формула кутової швидкості описується формулою 3, з формули швидкості колеса та кутової швидкості отримуємо рівняння лінійної швидкості 3.1:

$$\omega = \frac{2\pi}{T}, \quad (3)[11]$$

					ІТ61020БАК.004 ПЗ	Архив
Зм	Арк.	№ докum.	Підпис	Дата		47

$$v = \frac{l}{T}, \quad (3.1)$$

Якщо застосувати рівняння 3.1 до обох коліс, то формула буде однаковою, що для лівого, що для правого колеса, на (рис. 3.11) зображено детальну схему диференційного приводу.

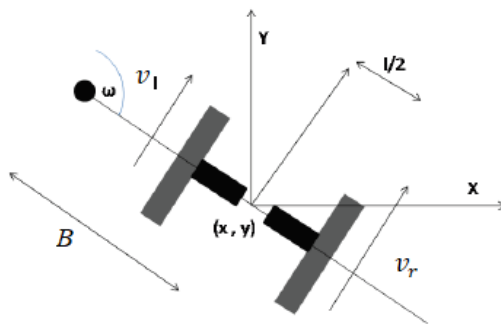


Рисунок 3.11 - Схема системи з диференційним приводом

Рівняння диференційного приводу коліс робота подано нижче, формули для правого та лівого коліс ідентичні

$$\omega \left(B + \frac{l}{2} \right) = v_{r,l}, \quad (3.2)$$

де B – відстань від центра кривої і центром осі, яка проходить через колеса робота, l – довжина осі колеса. Далі можемо застосувати ці формули в кодї програми. Розібравшись з кінематикою руху робота перейдемо до реалізації проєкту.

Для початку потрібно зрушити робота з місця, для отримання інформації з енкодерів та визначити швидкість з якою може пересуватись наш робот

					ІТ61020БАК.004 ПЗ	Арквш
Зм	Арк.	№ докum.	Підпис	Дата		48

використовуючи формули 3.2, 3.3, але дещо їх адаптувати до коду, далі вивести коефіцієнт, для перетворення команд користувача в сигнал на мотори.

Почнемо з команд користувача. Для цього нам потрібно визначити в який топик надсилаються команди та в якому форматі. В Росі деякі команди загально прийняті тому нам не потрібно довго шукати, команди які відповідають за рух платформи надсилаються в топик `cmd_vel`, тому в нашому модулі управління потрібно реалізувати підписку на цей топик та обробляти дані типу `geometry_msgs/Twist`, який містить в собі два вектори, які відповідають за лінійну швидкість та кутову швидкість (рис. 3.12), тип даних `Twist`.

```
$ rosmmsg show geometry_msgs/Twist
```

```
geometry_msgs/Vector3 linear
float64 x
float64 y
float64 z
geometry_msgs/Vector3 angular
float64 x
float64 y
float64 z
```

Рисунок 3.12 - Тип даних команди для руху

Також ROS має свій набір команд, для простішої взаємодії дана команда `rosmmsg` з ключем `show` відображає з чого складається тип `geometry_msgs/Twist`, і ми можемо швидко переглянути, які поля нам потрібно заповнити даними. Для передачі команди роботу нам потрібно вказати лінійну швидкість `linear.x` та кутову швидкість `angular.z`, тобто переміщення вздовж осі `x` та поворот навколо осі `z`. В Рос прийняті такі одиниці вимірювання, як лінійна швидкість в метрах на секунду, кутова швидкість в радіанах на секунду.

Коли ми запустимо програму, яка відсилає команди користувача до теми `cmd_vel` то в цій темі ми будемо отримувати повідомлення, які зображені на (рис. 3.13).

					IT61020БАК.004 ПЗ	Архив
Зм	Арк.	№ докum.	Підпис	Дата		49

```

linear:
  x: -0.988633466414
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: -1.2239008044
---
linear:
  x: -1.66241014919
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 1.11694186093
---
```

Рисунок 3.13 - Повідомлення теми cmd_vel

Як показано на (рис. 3.13), ми маємо два вектори, потрібно перетворити ці значення в сигнали на мотори робота. Якщо значення від'ємні відповідно в полі лінійної швидкості сигнал формується для руху назад і додатне відповідно вперед. В полі кутової швидкості від'ємне значення вказує на поворот праворуч і додатне відповідно ліворуч. Отже знаючи, що наш робот може рухатися з максимальною швидкістю 3 км/год, конвертуємо значення в 0.833 м/с, так як в Рос лінійна швидкість задається в м/с. Відповідно коли система отримує команду від користувача, команда перетворюється в ШІМ сигнал на мотори і робот починає свій рух. Далі щоб відстежувати рух робота потрібно отримувати дані про його переміщення та курсу з допомогою енкадерів та IMU датчика. Використовуючи імпульси енкадера ми отримуємо швидкість, а з IMU датчика кут на який робот змінює курс таким чином використовуючи наступні формули, можемо формувати одометрію.

$$\delta x = (v_x * \cos(\theta)) * \delta t, \quad (3.3)$$

$$\delta y = (v_x * \sin(\theta)) * \delta t, \quad (3.4)$$

$$\delta \varphi = \omega * \delta t, \quad (3.5)$$

					IT61020БАК.004 ПЗ	Архив
Зм	Арк.	№ докum.	Підпис	Дата		50

де v_x, v_y - лінійні швидкості по відповідним осям, φ – кут відхилення від курсу, ω – кутова швидкість робота та δt час між вимірами. З допомогою цих формул постійно обчислюється зміщення робота, тому потрібно його постійно фіксувати. Дані отримані з допомогою цих формул постійно записуються в координати, які відображають положення робота.

Після того як створено модуль одометрії, наша система може визначати своє положення в просторі, можна перейти до модуля навігації та побудови маршруту, хоч цей модуль є в бібліотеці Ros, він потребує налаштування під конкретну платформу. Цей модуль потребує відображення карти для визначення положення робота на ній та побудови маршруту. Побудова карти відбувається з використанням SLAM алгоритму. Під час руху робота, датчики сканують довкілля та одночасно йде зв'язування даних з датчиків до конкретних координат на карті, які ми отримуємо з одометрії, так ми отримуємо відображення карти, яка накладається на координатну площину робота.

SLAM – у обчислювальній геометрії алгоритм знаходження положення системи та побудови карти, це обчислювальна проблема побудови та побудови карти невідомого середовища при одночасному визначенні місця положення об'єкта в ньому.

Хоча це спочатку видається проблемою з курки та яйця, існує кілька алгоритмів, які дозволяють вирішити цю проблему, принаймні приблизно, в час спостереження для певного середовища. До популярних приблизних методів рішення відносяться фільтр частинок, розширений фільтр Кальмана, перетин коваріації та GraphSLAM.

Алгоритми SLAM використовуються в навігації, роботизованому картографуванні та одометрії, також для віртуальної або доповненої реальності, для побудови карти чи визначення положення об'єкта в просторі. На (рис. 3.14) зображено відображення карти в інструменті графічного відображення Rviz, яка побудована з використанням SLAM алгоритму [12].

						Адквш
Зм	Арк.	№ докум.	Підпис	Дата	IT61020БАК.004 ПЗ	51

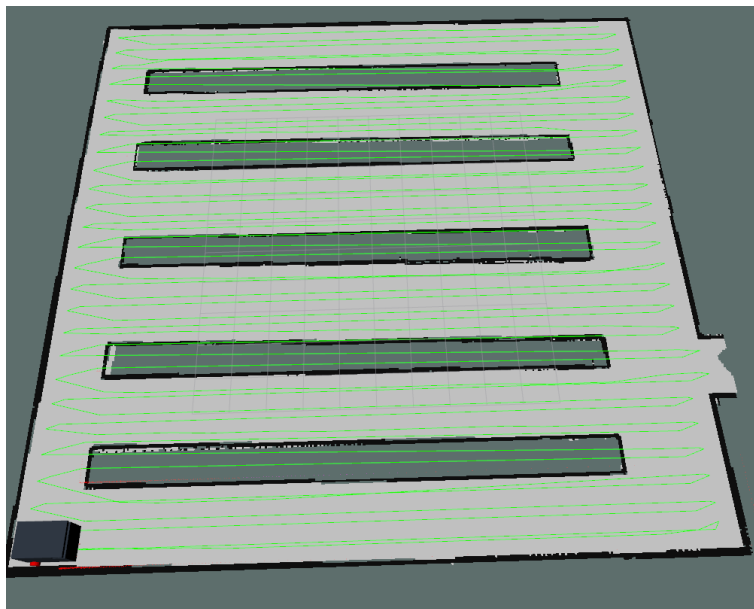


Рисунок 3.14 - Карта побудована алгоритмом SLAM

Для побудови відображення карти на робота потрібно встановити відповідний набір датчиків. Є декілька способів сканування довкілля та формування карти, найпоширеніший з допомогою лазерного далекоміра (рис. 3.15), проте також можна використовувати камеру глибини або стерео камеру.

Звісно вигідніше використовувати стерео камеру, так як вона може давати дані подібні до даних лазерного далекоміру, навіть кращі. Дає можливість використовувати її як звичайну камеру та використовувати дані в машинному зорі, що значно розширює можливості системи не тільки будувати карту та орієнтуватися на ній, а й мати можливість розпізнавати різні об'єкти та дає можливість реалізувати більш складну логіку керування.

						ІТ61020БАК.004 ПЗ	Архив
Зм	Арк.	№ докum.	Підпис	Дата			52



Рисунок 3.15 - Лазерний далекомір [12]

Для того, щоб карта будувалася правильно, потрібно описати положення кожного датчика, який використовується для побудови карти, від того як точно будуть задані положення датчиків залежить якість карти. В росі для цього використовується бібліотека трансформацій, яка відповідно до ієрархії частин робота з допомогою матриць повороту змінює положення кожної частини, якщо це потрібно. Відлік відстані починається з фізичного центру робота та задається в метрах. Після проведення всіх маніпуляцій можемо запустити систему та відобразити поточний стан робота в Rviz і побачити як робот сприймає довколишнє середовище (рис. 3.16).

На (рис. 3.16) відображається робот в програмі візуалізації Rviz, зелена, червона та рожева лінії відображають дані з лазерного далекоміра, у вигляді конусів відображаються дані з ультразвукових датчиків, з допомогою цього інструменту ми можемо відразу відстежувати зміни в сприйнятті робота та коригувати його стан.

						ІТ61020БАК.004 ПЗ	Аркуш
Зм	Арк.	№ докum.	Підпис	Дата			53

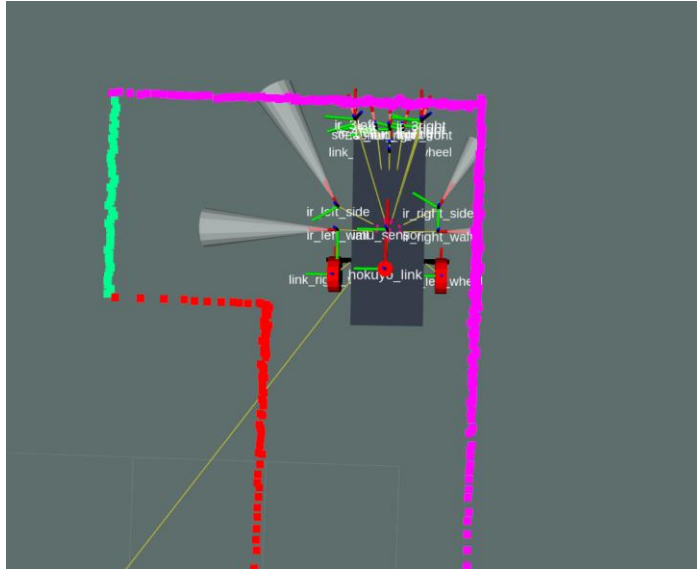


Рисунок 3.16 – Відображення робота в Rviz

Використовуючи графічний інструмент Rviz, протестовані основні компоненти, які потрібні для побудови карти: лазерний далекомір, одометрія, та ІМУ-датчик. Тепер потрібно запустити всі модулі разом, щоб вони могли передавати між собою необхідні дані.

Після успішного запуску системи, можемо запустити модуль, який збирає всі дані з різних джерел, утворює з них координатну площину, та до кожної координати будує відношення зі скануванням лазерного далекоміру.

Для тестування модуля картографування в симуляторі потрібно побудувати тестове середовище з фіксованими перешкодами, зображення робота в симуляторі подано на (рис. 3.17) та можна відкрити Rviz, для перегляду роботи системи в реальному часі і мати змогу відловити проблеми та виправити їх, відповідно відображення даних та процес картографування на (рис. 3.18).

						ІТ61020БАК.004 ПЗ	Адквш
Зм	Арк.	№ докум.	Підпис	Дата			54

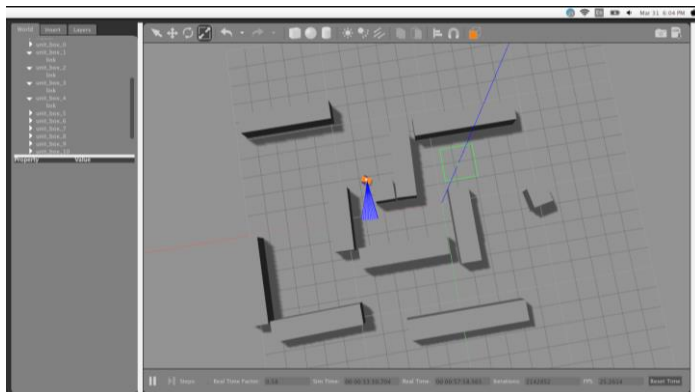


Рисунок 3.17 – Тестове середовище в симуляторі



Рисунок 3.18 – Rviz процес побудови карти

Як ми бачимо на (рис. 3.18) побудова карти відбувається правильно без спотворень, всі об'єкти середовища добре відображаються та всі на своїх місцях, з цього можемо зробити висновок, що створені модулі працюють

					IT61020БАК.004 ПЗ	Адквш 55
Зм	Арк.	№ доквм.	Підпис	Дата		

добре, залишився останній пункт, це задання точки в середовищі та перевірка роботи системи в процесі побудови маршруту та переміщення по цьому маршруту.

Навігаційний модуль включає в себе реалізацію дій, які беруть керування переміщенням на себе та намагаються досягти заданої цілі в просторі [13]. Даний модуль включає в себе планування маршруту як в локальній області довкола платформи так і в глобальному просторі та надає інтерфейс ROS для налаштування, виконання та взаємодії навігаційного стеку на самому роботі.

Діаграма розгортання системи наведена на кресленику IT61.020БАК.004 ДЗ, де зображено взаємодію головного хоста на якому запущений основний процес, який зв'язує між собою всі вузли та утворює мережу та схему роботи з модулями, які виконують обчислення під час роботи різних режимів. Для правильної роботи навігаційного модуля в файлі конфігурації платформи, потрібно заповнити дані про робота, його розміри, швидкість та налаштування для формування команд для правильної роботи системи, уникання зіткнень та планування шляху.

Під час тестування системи, виявлені деякі недоліки, за певний час переміщень накопичення помилки починає впливати на систему і в деякий момент, система починає працювати некоректно, тому було прийнято рішення розглянути спосіб тріангуляції з використанням маяків.

Тріангуляція відноситься до рішення рівнянь з обмеженнями пов'язаних з позицією спостерігача до позицій набору орієнтирів. Визначення місцеположення з використанням методу тріангуляції та відомих орієнтирів почали використовуватися дуже давно, ще стародавніми римлянами для картографування та будівництва доріг за часів Римської імперії. Найпростіший і найвідоміший випадок, який дає назву цьому способу – це знаходження напрямку до орієнтиру або вимірювання відстані двох (або більше) орієнтирів для вирішення задачі планарного позиціонування, таким

					IT61020БАК.004 ПЗ	Аркvш
Зм	Арк.	№ докum.	Підпис	Дата		56

чином використовуючи трикутну схему розташування орієнтирів, дає комбінацію сторін та кутів. Цей метод знаходження місцеположення сягає своїм корінням ще в давнину в контексті архітектури та картографії і є важливим сьогодні у кількох галузях, таких як наука про визначення місця в просторі. Хоча розміщення орієнтирів по трикутній схемі не єдина можлива конфігурація, для використання орієнтирів або маяків, вона є найбільш практичною [13]. Хоча орієнтири, маяки та роботи існують у тривимірному світі, обмеження точності пов'язане з інформацією про висоту, часто на практиці це проблема в двовимірному просторі. Інформація про висоту іноді використовується для підтвердження результатів. Таким чином, хоча проблему триангуляції для визначення положення робота слід розглядати як проблему із шістьма невідомими параметри (три змінні положення та три змінні орієнтації), частіше завдання ставиться як двовимірне (або тривимірне) задача з двовимірною орієнтири [14].

Залежно від комбінацій сторін (S) та кутів (A) задана проблема триангуляції описується як (SAS) тощо. Усі випадки дозволяють вирішити завдання за винятком випадку AAA, у якому трикутник не обмежений параметрами.

На практиці, дана технологія зондування часто повертає обидва значення кута або відстані і розташування орієнтира зазвичай відоме.

Таким чином, SAA і випадки SSS є найбільш поширеними. Загалом, проблема може включати певну комбінацію алгебраїчних обмежень, що стосуються вимірювань до розташування орієнтирів. Вони, як правило, нелінійні, отже рішення може залежати від вихідної оцінки положення або обмеження [16].

Якщо доступна лише відстань до орієнтирів, одиничне вимірювання обмежує положення робота до дуги кола. на (рис. 3.16) зображено простий спосіб триангуляції.

					IT61020БАК.004 ПЗ	Аркуш
Зм	Арк.	№ докum.	Підпис	Дата		57

Робот у невідомому місці x_1 помічає два маяки P_1 і P_2 , вимірюючи відстані d_1 і d_2 до них. Це відповідає нашому випадку, в якому маяки у відомих місцях активно випромінюють сигнал і робот отримує відстані на основі часу за який сигнал надходить до робота. Робот повинен знаходитися на перетині кола радіуса d_1 з центром у P_1 та з колом чи радіусом d_2 з центром на P_2 .

Без втрати загальності можна припустити, що P_1 знаходиться на початку і що P_2 знаходиться на $(h, 0)$. Тоді маємо рівняння 3.9.

$$d_1^2 = x^2 + y^2, \quad (3.6) [16]$$

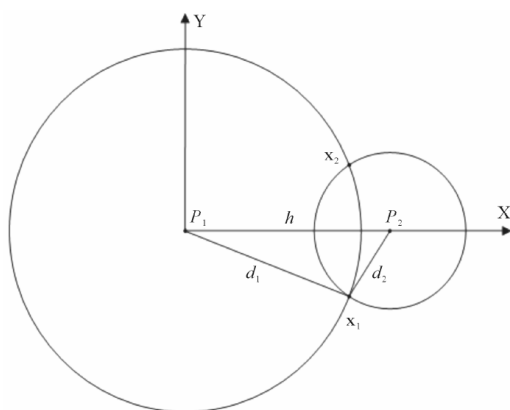


Рисунок 3.16 – Приклад тріангуляції, визначення невідомої координати робота x_1

$$d_2^2 = (h - x)^2 + y^2, \quad (3.7) [16]$$

Де h - відстань між двома точками, d_1 і d_2 – відповідно радіуси кіл, які перетнулися в точці x_1 . Застосувавши деякі алгебраїчні перетворення отримаємо

$$x = \frac{h^2 + d_1^2 - d_2^2}{2h}, \quad (3.8) [16]$$

$$y = \pm \sqrt{d_1^2 - x^2} = \pm \sqrt{d_1^2 - \left[\frac{h^2 + d_1^2 - d_2^2}{2h} \right]^2}, \quad (3.9) [16]$$

в результаті виходить два рішення для знаходження x_1 і x_2 . У типовому застосуванні маяки розташовані на стінах і таким чином хибне (у нашому прикладі, x_2) рішення може бути ідентифіковане, оскільки воно вказує на некоректне розташування робота, в середині стіни. Хоча відстані до маяків надають простий приклад тріангуляції, більшість датчиків і орієнтирів призводять до більш складних ситуацій [17].

Судячи з досліджень даного способу визначення положення об'єкта на місцевості з використанням маяків, може покращити точність даних. Але даний спосіб має обмеження в розмірі карти, тобто розмір місцевості на якій може відбуватися визначення розташування об'єкта обмежується потужністю передавачів. На (рис. 3.19) зображено порівняння двох способів визначення положення об'єкта на місцевості.

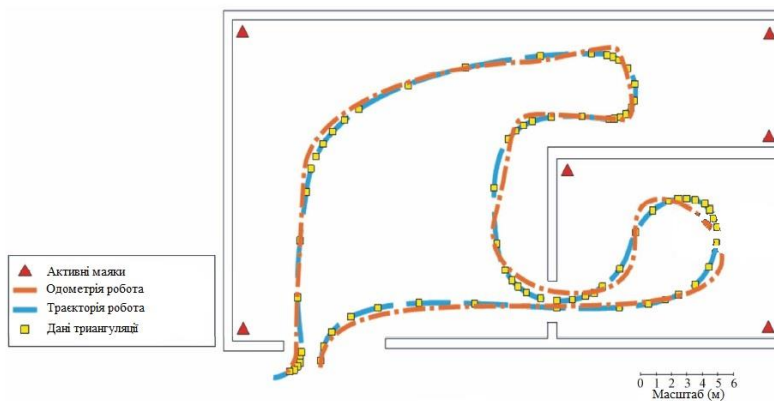


Рисунок 3.19 – Навігація робота в декількох кімнатах

Використання ToF радіочастотного сигналу, що взаємодіє з RF маяками для обчислювальної тріангуляції для забезпечення визначення положення в двовимірному просторі в приміщенні було підтверджено з тими першими експериментальними роботами. Це відносно дешева система впровадження, яка надає хороші результати для навігації мобільного робота в приміщенні, де система GPS не має змоги працювати. Таким чином, метод тріангуляції з використанням маяків надає досконалу альтернативу для традиційної ультразвукової навігаційної технології, з низькими витратами на реалізацію RF системи цифрового приймача з використанням маяків.

Дані датчиків, такі як одометрія, компас і результат тріангуляції визначення положення, були скомпоновані з допомогою фільтра Калмана таким чином, щоб отримати оптимальні дані та правильну позицію робота.

Серед інших параметрів які були спрямовані на реалістичну модель накопичення помилки в визначенні положення відносно маяка та відстані між роботом і вона збільшується.

Результати моделювання, пов'язані з експериментальними перевірками підтверджують, що ця методика є правильною і багатообіцяючою, що мобільні роботи в автономному режимі можуть бути в змозі корегувати під час руху власну траєкторію. Консистенція даних, що комбінуються відповідно одометрії та дані з компасу мобільного робота і результат RF тріангуляції маяків не спотворюються навіть після збурень в системі.

Представлений метод не робить миттєву абсолютну локалізацію досконалою, але успішні послідовні вимірювання показують, що визначення положення з допомогою цього методу найкраще впливає на роботу по знаходженню системи на місцевості.

					IT61020БАК.004 ПЗ	Аркvш
Зм	Арк.	№ докum.	Підпис	Дата		60

Висновок до розділу

Після реалізації даного проєкту можна зробити висновки, що побудова роботизованих систем досить непроста задача, так як існує багато нюансів, які залежать від навколишнього середовища та фізичних факторів, які впливають на роботу системи.

Але такий інструмент як Ros, дозволяє спростити деякі частини розробки та дає можливість спростити розробку роботизованих систем взявши на себе реалізацію загально прийнятих систем та сконцентрувати увагу розробника саме на реалізації власних алгоритмів і адаптації системи під конкретне місце використання системи. Хоча Ros і допомагає з деякими аспектами розробки, але фізичні фактори потрібно вирішувати самостійно, такі як калібрування та фіксування похибок в датчиках, фільтрування їх даних для зменшення похибок по всій системі в цілому. Під час виконання проєкту, експериментальним шляхом було виявлено недоліки при використанні датчиків низької якості та використання в системі одного джерела інформації з датчиків, було прийняте рішення дублювати необхідні системи, такі як IMU датчики, енкодери та далекоміри. Потім дані отримані від цих систем комбінувати в один масив та використовувати його для орієнтації, таким чином можна виключити перешкоди в сигналах датчиків та розбіжність значень. Проте під час тестування системи, було виявлено недолік, який впливає на точність системи – накопичення похибки. Тому було прийнято рішення застосувати до вже готової системи ще один спосіб знаходження місцеположення на площині з допомогою використання маяків для тріангуляції.

Даний спосіб експериментально підтвердив свою здатність надавати правильні та точні дані. З використанням цього способу точність системи покращилась, до того ж вона може працювати довгий час без перебоїв, що дуже важливо в керуванні мобільними роботами.

					ІТ61020БАК.004 ПЗ	Аркvш
Зм	Арк.	№ докum.	Підпис	Дата		61

ВИСНОВКИ

В результаті роботи над проєктом було створено систему керування мобільними роботами та їх навігації, під час розробки було досліджено багато способів, для реалізації навігації, та всі вони різні і використовуються відповідно до їх можливостей. Так як система повинна вільно пересуватися та мати змогу самостійно керувати своїм переміщенням або з допомогою оператора, було прийняте рішення, обрати спосіб який базується на одометрії.

Після реалізації та проведення тестування працездатності системи, було виявлено ряд недоліків даного підходу, під час довгочасного переміщення починає накопичуватися похибка, яка далі впливає на точність позиціонування системи.

Основуючись на цих даних було вирішено продублювати датчики, які відповідають за надання інформації для позиціонування системи, це покращило результати, але на практиці не існує ідеальних систем, навіть калібрування та поправки на похибки не дають найкращих даних. Тому було прийняте рішення спробувати спосіб тріангуляції з використанням маяків, після імплементації даного способу та комбінування його даних з даними попередніх систем, система покращила свої результати при позиціонуванні на місцевості.

Завдяки реалізації проєкту з використанням інструменту ROS його можна легко переносити на інші системи, з мінімальними змінами. Та може бути розширена доповнена новими способами знаходження місцеположення в різних середовищах відповідно до вимог.

					ІТ61020БАК.004 ПЗ	Архив
Зм	Арк.	№ докum.	Підпис	Дата		62

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Инструментарии роботостроения [Электронный ресурс] / Колорадо, М. Тим Джонс. - Режим доступа до ресурсу: [www/ URL: http://developerworks/ ru/library/lrobotools/#author.html/](http://developerworks.ru/library/lrobotools/#author.html/) - 05.09.2008 г. - Загл. з екрану.
2. Навигация зрячего робота [Электронный ресурс] Режим доступа до ресурсу: [www/ URL: http://cofelis.ru/?page_id=46&page=3.html/](http://cofelis.ru/?page_id=46&page=3.html/) - 17.09.2008 г. - Загл. з екрану.
3. Навигация мобильных роботов [Электронный ресурс] / Лондон, Имперский Колледж. - Режим доступа: [www/ URL: http://computer-vision.ucoz.ru/MobRoboNavigati/mobile_robot_navigation.html/](http://computer-vision.ucoz.ru/MobRoboNavigati/mobile_robot_navigation.html/) - 10.06.1997 г. - Загл. з екрану.
4. Бобровский, С.Н. Навигация мобильных роботов [Текст] / С.Н. Гончаров// Журн. PC Week. - 2004. - №9. - С. 60-63
5. Мартыненко, Ю. Г. Управление движением мобильных колёсных роботов [Текст] / Ю.Г. Мартыненко - МГУ им. М.В. Ломоносова, 2005. - 29-80с.
6. An indoor mobile robot navigation technique using odometry and electronic compass [Электронный ресурс]. – 2017. – Режим доступа до ресурсу: <https://journals.sagepub.com/doi/full/10.1177/1729881417711643>.
7. Building a Magnetic Track Guided AGV [Электронный ресурс]. – 2017. – Режим доступа до ресурсу: <https://www.roboteq.com/applications/all-blogs/18-building-a-magnetic-track-guided-agv>.
8. ROS/Concepts [Электронный ресурс]. – 2014. – Режим доступа до ресурсу: <http://wiki.ros.org/ROS/Concepts>.
9. Joint element description [Электронный ресурс]. – 2014. – Режим доступа до ресурсу: <http://wiki.ros.org/urdf/XML/joint>.

						IT61020БАК.004 ПЗ	Арквш
Зм	Арк.	№ докum.	Підпис	Дата			63

10. SIEGWART R. Mobile Robot Kinematics / R. SIEGWART, I. NOURBAKHSI // Introduction to Autonomous Mobile Robots / R. SIEGWART, I. NOURBAKHSI., 2004. – (Massachusetts Institute of Technology). – С. 47–82.

11. Лентин Д. Изучение робототехники с использованием Python / Джозеф Лентин. – Москва: ДМК Пресс, 2019. – 250 с. – (ДМК Пресс).

12. SLAM algorithm applied to robotics assistance for navigation in unknown environments [Электронный ресурс] // Journal of NeuroEngineering and Rehabilitation. – 2010. – Режим доступа до ресурсу: <https://jneuroengrehab.biomedcentral.com/articles/10.1186/1743-0003-7-10#Sec12>.

13. Navigation Stack move_base [Электронный ресурс]. – 2018. – Режим доступа до ресурсу: http://wiki.ros.org/move_base.

14. Dudek G. Computational Principles of Mobile Robotics / G. Dudek, M. Jenkin. – London: Cambridge University Press, 2000. – 406 с.

15. Gonzalez-Banos H. Robot Navigation for Automatic Model Construction Using Safe Regions / H. Gonzalez-Banos, J.-C. Latombe // Lecture Notes in Control and Information Sciences / H. Gonzalez-Banos, J.-C. Latombe. – Berlin: Springer, 2001. – С. 405–415.

16. Kam M. Sensor Fusion for Mobile Robot Navigation / M. Kam, X. Zhu, P. Kalata // Proceedings of the IEEE Vol. 85, Issue: 1 / M. Kam, X. Zhu, P. Kalata., 1997. – С. 108–119.

17. Tikhonov A. N. Solutions of Ill-posed Problems / A. N. Tikhonov, V. Y. Arsenin., 1977. – 258 с. – (the University of Michigan).

18. Thielscher M. Reasoning Robots The Art and Science of Programming Robotic Agents / Thielscher. – Netherlands: Springer, 2005

					IT61020БАК.004 ПЗ	Адквш
Зм	Арк.	№ докум.	Підпис	Дата		64